IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Swander et al.	Examiner:	Williams, Jeffery L.
Serial No.:	10/056889	Group Art Unit:	2137
Filed:	01/25/2002	Docket No.:	14917.0431US01 (formerly M1103.70145US00)
Title:	METHOD AND APPARATUS FOR FRAGMENTING AND REASSEMBLING INTERNET KEY EXCHANGE DATA PACKETS		

CERTIFICATE UNDER 37 CFR 1.8:

I hereby certify that this document is being transmitted electronically to the U.S. Patent Office on June 7, 2007.

Jenifer Weck

AMENDMENT

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Office Action mailed December 8, 2006, please amend the above-identified application as follows:

Amendments to the Claims begin on page 2 of this paper.

Remarks/Arguments begin on page 8 of this paper.

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application. Please amend the claims as follows:

Listing of Claims:

1. (Original) A method for transmitting Internet Key Exchange (IKE) data packets across a network comprising the steps of:

generating and transmitting an IKE packet over a network;

determining whether a response to the IKE packet was received;

fragmenting the IKE packet into a plurality of smaller packets when a response is not received, wherein each of the smaller packets includes a header formatted according to the IKE protocol; and

transmitting each of the plurality of smaller packets over a network.

2. (Original) The method of claim 1 wherein each header includes an identifier that may be used to associate the smaller packet with a corresponding IKE packet.

3. (Previously presented) A network node that communicates with other network nodes according to the Internet Key Exchange (IKE) protocol comprising:

a User Datagram Protocol (UDP) stack that is capable of generating UDP data packets for transmission over a network;

an IKE protocol stack that generates IKE data packets that are subsequently processed by the UDP protocol stack; and

a fragmenter module that intercepts IKE data packets prior to being processed by the

UDP protocol stack and splits the IKE data packets into a plurality of smaller data packets that may be subsequently formatted by the UDP protocol stack;

wherein the fragmenter module does not split the IKE data packets unless no response to a previously-sent IKE data packet has been received; and

wherein each of the plurality of smaller data packets includes a header formatted according to the IKE protocol.

4. (Canceled)

5. (Canceled)

6. (Previously presented) A method for receiving fragmented Internet Key Exchange (IKE) data packets comprising the steps of:

receiving a plurality of fragments of an IKE data packet from a transmitting node, wherein each fragment includes an identifier that associates each fragment with an IKE data packet;

discarding all fragments that contain a first identifier if a predetermined number of fragments are received that contain a second identifier; and

determining the total size of all fragments that contain the same identifier and discarding said fragments when the total size exceeds a predetermined limit.

7. (Original) The method according to claim 6 wherein the step of discarding all fragments that contain a first identifier is performed when at least one fragment is received that contains a second identifier.

8. (Original) The method according to claim 6 further comprising the steps of:

determining whether all fragments that are associated with an IKE data packet have been

received; and

sending a no acknowledgment (NAK) message to the transmitting node when at least one fragment has not been received.

9. (Canceled)

10. (Previously presented) The method according to claim 6 wherein the predetermined limit is 64 kilobytes.

11. (Previously presented) A system for transmitting Internet Key Exchange (IKE) protocol data packets across a network comprising:

means for generating an IKE packet;

means for initializing, operating, and monitoring a timer;

means for detecting whether the IKE packet was successfully received at the intended receiver node before the expiration of the timer; and

means for fragmenting the IKE packets into smaller packets when the IKE packet was not successfully received at the receiver node before the expiration of the timer, wherein each of the smaller packets includes information that permits a receiver node to identify the IKE packet associated with each smaller packet and the position of each smaller packet within the IKE packet.

12. (Original) The system of claim 11 further comprising means for determining the capability of the receiver node for receiving fragmented packets.

13. (Original) A method for transmitting data packets across a network comprising the steps of:

generating and transmitting an Internet Key Exchange (IKE) packet over a network;

determining whether a response to the IKE packet was received;

fragmenting the IKE packet into a plurality of smaller packets when a response is not received; and

transmitting each of the plurality of smaller packets over a network.

14. (Previously presented) The method of claim 13 wherein each of the plurality of smaller packets contains a header formatted according to the IKE protocol.

15. (Previously Presented) The method of claim 13 wherein the IKE packet contains a header formatted according to the IKE protocol.

16. (Previously Presented) The method of claim 15 wherein the plurality of smaller packets contain the same information as that contained within the original IKE packet.

17. (Previously Presented) The method of claim 16 wherein at least one of the plurality of smaller packets contains the header formatted according to the IKE protocol.

18. (Previously presented) A method for transmitting data packets across a network comprising the steps of:

generating a data packet containing Internet Key Exchange (IKE) information;

initializing a timer;

determining, based at least in part on the expiration of the timer, whether fragmentation of the data packet is necessary to successfully transmit the IKE information over a network; and

fragmenting the data packet if necessary into a plurality of smaller packets that may be transmitted over a network.

19. (Canceled)

20. (Previously presented) A method for resolving transmitting errors

associated with transmitting Inter Key Exchange (IKE) packets via protocol stacks that implement the Transmission Control Protocol (TCP), the User Datagram Protocol (UDP), and/or the Internet Protocol (IP) comprising the steps of:

generating a data packet containing IKE data;

initializing a timer;

determining, based at least in part on the expiration of the timer, whether it is necessary to fragment the IKE data packet;

fragmenting the packet, if necessary, with a code module that does not implement the TCP, UDP, or IP protocols before the packet is processed by a code module that does implement the TCP, UDP or IP protocols; and

transmitting the fragmented packet over a network.

21. (Canceled)

22. (Previously presented) A method for intelligently discarding fragmented Internet Key Exchange (IKE) data packets to efficiently manage resources comprising:

receiving a plurality of fragments of a single IKE data packet, wherein the fragments were transmitted from a transmitting node in an order that can be determined from information contained within the received fragments;

determining from information contained within the received fragments whether any of the received fragments have been received in an order that differs from the order in which the fragments were transmitted from the transmitting node; and

discarding at least certain of the received fragments when a predetermined number of out of order fragments from a single IKE data packet have been received.

23. (Previously Presented) The method of claim 22 further including the step of sending a message to the transmitting node that out of order packets have been received.

REMARKS/ARGUMENTS

This Amendment and the following remarks are intended to fully respond to the Office Action mailed December 8, 2006. In that Office Action claims 1-23 were examined, and all claims were rejected. More specifically, the specification is objected to as failing to provide proper antecedent basis for the claimed subject matter; claim 3 was rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement; claims 1-3, 6-8, 10-18, 20, 22, and 23 were rejected under 35 U.S.C. § 103(a) as being unpatentable over IPSEC, "Minutes of IPSEC Working Group Meeting," hereinafter "IPSEC Minutes," in view of Kent et al., "Fragmentation Considered Harmful," hereinafter "Kent;" and claims 22 and 23 were rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of IPSEC Minutes and Kent in view of Cert et al., "A Protocol for Packet Network Intercommunication," hereinafter "Cert." Reconsideration of these rejections, as they might apply to the original and amended claims in view of these remarks, is respectfully requested.

In this Response, no claims have been amended, canceled, or added.

Specification

The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter. The examiner states:

> Claim 3 comprises the limitation "*wherein the fragmenter module does not split the IKE data packets unless no response to a previously-sent IKE data packet has been received.*" The applicant has not pointed out where the amended claim is supported, nor does there appear to be a written description of the claim limitation in the application as filed.
> 12/8/2006 Office Action, p. 2, "Specification."

Applicants wish to direct the Examiner's attention to both Fig. 18 and the corresponding

description found in the Specification from page 18, line 15 to page 19, line 22. At least one embodiment that describes the above claim limitation is presented in that potion of the specification.

Claim Rejections – 35 U.S.C. § 112

Claim 3 was rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement.

Again, Applicants wish to direct the Examiner's attention to both Fig. 18 and the corresponding description found in the Specification from page 18, line 15 to page 19, line 22. At least one embodiment that describes the above claim limitation is presented in that potion of the specification. In embodiments and as described in the Specification, "[t]he fragmenter 150 monitors the message and, after a suitable time interval, determines whether appropriate response has been received. If a suitable response has been received, the fragmenter 150 will simply permit the process to continue according to the usual IKE protocol, step 183. According to this logic, no fragmentation will occur if the IKE payloads are successfully transmitted." Specification, Page 19, lines 2-7.

Claim Rejections – 35 U.S.C. § 103

Claims 1-3, 6-8, 10-18, 20, 22, and 23 were rejected under 35 U.S.C. § 103(a) as being unpatentable over IPSEC Minutes in view of Kent.

Under the MPEP 715.07, Applicants should show one of the following to swear behind a reference:

> (a) reduction to practice of the invention prior to the effective date of the reference; OR

> (b) conception of the invention prior to the effective date of the reference coupled with due diligence from prior to the reference date to a subsequent actual reduction of practice; OR
>
> (c) conception of the invention prior the effective date of the reference coupled with due diligence from prior to the reference date to the filing of the patent application.

Applicants hereby submit a 37 CFR 1.131 declaration (Exhibit A) and supporting attachments (Exhibits B, C, and D) demonstrating that the present invention was conceived and reduced to practice prior to the December 12, 2001 effective date of the IPSEC Minutes. Exhibit B, attached hereto, contains an email from inventor Brian Swander. Exhibit C, attached hereto, is an attached document to the email titled ikefrag.doc, referred to as the Description Document, which describes the invention, i.e., the ISA_FRAG payload, and the efforts to reduce the invention to practice. Finally, Exhibit D is a document titled Microsoft Patent Predisclosure Document.

The Description Document in Exhibit C describes the software system including the operation of the ISA_FRAG payload, which was previously conceived and was reduced to practice. The ISA_FRAG payload is the fragmented IKE payload that is sent to a receiver and reassembled. Exhibit C documents the early conception and shows that a working embodiment was coded and tested. Applicants believe this showing is sufficient evidence to demonstrate that there was an actual reduction to practice on or before December 5, 2001 (see Corona M. Dovan, 273 U.S. 692, 1928 C.D. 252 (1928) "A process is reduced to practice when it is successfully performed"), when the claimed methods were performed by the software being tested as documented in Exhibit C.

The Description Document is evidence of conception and reduction to practice of the claimed invention prior to December 12, 2001. Applicants believe the document shows detail of

sufficient character and weight as required under 37 CFR 1.131(b) to establish that the present invention as claimed was conceived and reduced to practice at least prior to the IPSEC Minutes reference's effective date of December 12, 2001.

Further, Applicants would also like to draw Examiner's attention to MPEP § 716, wherein an Applicant may submit evidence of non-obviousness. For example, "it is 'well-settled' law that an inventor's own disclosure 'will not anticipate his later invention unless that prior work is such as to constitute a statutory bar under Section 102(b)." Lacks Indus. v. McKechnie Vehicle Components USA, Inc., 322 F.3d 1335, 1346 (Fed. Cir. 2003), quoting Donald S. Chisum, 1 Chisum On Patents § 3.08[2][a] (1999).

Applicants hereby submit a 37 CFR 1.132 declaration (Exhibit 1) and supporting attachments (Exhibits 2, 3, and 4) that the disclosure in the IPSEC Minutes reference derived from the inventors. Exhibits 2, 3, and 4, attached hereto, contain respectively a copy of the IPSEC Minutes for the IPSEC Working Group Meeting of December 12, 2001, a presentation entitled "IPSEC over NAT Testing" given by William Dixon at the IPSEC Working Group Meeting of December 12, 2001, and an affidavit from William Dixon that any information regarding IKE fragmentation presented at the IPSEC Working Group Meeting of December 12, 2001 derived from the inventors. These documents establish that the IPSEC Minutes cannot be used as a reference against the present application because the information contained therein derived from the inventors.

As the IPSEC Minutes reference cited by the Examiner is not prior art to the pending application, the Examiner has not made out a *prima facie* case of obviousness. Therefore,

Applicants respectfully request that the Examiner withdraw this rejection and find the claims in a condition for allowance.

Claims 22 and 23 were rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of IPSEC and Kent in view of Cert. For the same reasons as above, the IPSEC reference is not prior art. Therefore, Applicants respectfully request that the Examiner withdraw this rejection and find the claims in a condition for allowance.

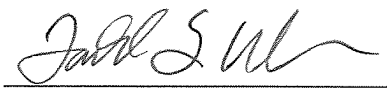
Conclusion

It is believed that no further fees are due with this Response. However, the Commissioner is hereby authorized to charge any deficiencies or credit any overpayment with respect to this patent application to deposit account number 13-2725.

In light of the above remarks and amendments, it is believed that the application is now in condition for allowance and such action is respectfully requested. Should any additional issues need to be resolved, the Examiner is requested to telephone the undersigned to attempt to resolve those issues.

Respectfully submitted,

Dated: June 7, 2007



Tadd F. Wilson, #54,544
MERCHANT & GOULD P.C.
P.O. Box 2903
Minneapolis, MN 55402-0903
303.357.1651

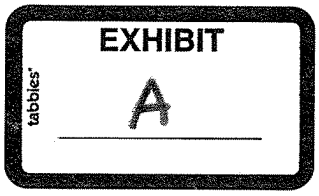


EXHIBIT
tabbies
A

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Swander, et al.	Examiner:	Jeffery L. Williams
Serial No.:	10/056,889	Group Art Unit:	2137
Filed:	January 25, 2002	Docket No.:	14917.0431US01
Title:	METHOD AND APPARATUS FOR FRAGMENTING AND REASSEMBLING INTERNET KEY EXCHANGE PACKETS		

DECLARATION OF BRIAN SWANDER & CHRISTIAN HUITEMA

PURSUANT TO 37 CFR 1.131

We, Brian Swander and Christian Huitema, hereby declare as follows:

1. We are joint inventors named on U.S. Patent Application Serial No. 10/056,889, filed January 25, 2002 (hereinafter, "the present application").

2. We are co-inventors of the subject matter disclosed and claimed in the present application.

3. We are aware of the Office Action in the present application mailed December 8, 2006 in which Examiner Jeffery L. Williams maintained "Minutes of IPSEC Working Group Meeting" (hereinafter, "IPSEC Minutes"), dated December 12, 2001, in view of "Fragmentation Considered Harmful" to Kent as a basis for rejecting independent claim 1 of the present application under 35 USC § 103(a).

4. We are aware of an Office Action Response and an Amendment in the present application being filed in response to the Office Action and that this declaration is attached to the Amendment as part of Exhibit A.

5. We are aware that to antedate a prior art reference, we must show that we both conceived the invention and reduced the invention to practice before the date of the prior art reference. We are also aware that to show a reduction to practice we must prove that we constructed an embodiment of the invention that met every element of the claims and that the embodiment operated for its intended purpose, or, alternatively show that the invention was constructively reduced to practice by filing the patent application. We hereby aver that the invention set forth in rejected claim 1 in the present application was conceived and reduced to practice by we the

inventors listed on the present application in this country at least as early as December 5, 2001 which is earlier than the filing date of the IPSEC Minutes reference.

a. Conception: Exhibit B, attached hereto, contains an email dated December 5, 2001, hereinafter referred to as "the email," with an attached electronic document titled "ikefrag.doc" (hereinafter the "Description Document"). The Description Document, attached to this affidavit as Exhibit C, describes the software system including sending IKE packets, determining if a response was received, fragmenting the IKE packets, and sending the fragments. The Description Document introduces a new IKE payload (ISA_FRAG) for sending IKE fragments. The Description Document was provided to the patent attorneys, in the Microsoft Patent Predisclosure Document shown in Exhibit D, to describe the invention and to allow the patent attorneys to draft the present patent application. As such, all subject matter of the claims in the present application were gleaned from the information provided in the Description Document. We will explain the presence of each element of claim 1 in the Description Document:

Claim 1 comprises: a *method for transmitting Internet Key Exchange (IKE) data packets across a network.* The method has four elements, including *generating and transmitting an IKE packet*; *determining whether a response to the IKE packet was received*; *fragmenting the IKE when a response is not received*; *wherein each of the smaller packets includes a header formatted according to the IKE protocol*; and *transmitting each of the plurality of smaller packets.*

1. The *method for transmitting Internet Key Exchange (IKE) data packets across a network* is mentioned in the "Solution" section. See Description Document, pg. 1, § **Solution** ("*incorporate limited fragmentation/*reassembly and MTU discoverability via black hole detection *into IKE.*") (emphasis added).

2. The *generating and transmitting an IKE packet* is also mentioned. See Description Document, pg. 1, § **Big Picture** ("*Send big IP payload as normal.* Wait for response...") (emphasis added).

3. The *determining whether a response to the IKE packet was received* is also mentioned. See Description Document, pg. 1, § **Big**

Picture ("*Wait for response.* Retransmit once or twice to allow for mormal lost packets, and slow peer validation. *Then, begin black hole detection.*") (emphasis added).

4. The *fragmenting the IKE when a response is not received; wherein each of the smaller packets includes a header formatted according to the IKE protocol* is also mentioned. See Description Document, pg. 1, § **Implementation** ("*If we are doing fragmentation*, we take the normal payload:

IP UDP IKEHDR [ID CERT SIG] where [] denoted encrypted

And send:

IP UDP IKE HDER FRAG1

IP UDP IKEHDR FRAG 2 . . .") (emphasis added). The smaller packets each have an IKE header. See Description Document, pg. 1, § **Implementation** ("IKEHDR").

5. Finally, the *transmitting each of the plurality of smaller packets* is also mentioned. See Description Document, pg. 1, § **Implementation** ("If we are doing fragmentation, we take the normal payload:

IP UDP IKEHDR [ID CERT SIG] where [] denoted encrypted

And send:

IP UDP IKE HDER FRAG1

IP UDP IKEHDR FRAG 2 . . .") (emphasis added).

ii. The other independent claims, claims 3, 6, 11, 13, 18, 20, and 22 have similar limitations

iii. As the preceding demonstrates, the elements of the claims were conceived of and documented in the Description Document. The conception of the invention occurred on or before December 5, 2001, and the Description Document documented and described the invention as conceived on December 5, 2001. The patent application claims the invention as described in the Description Document.

b. Reduction to Practice: Exhibit B and C attached hereto, contain respectively information about the prototype of the software, ISA_FRAG, and the testing of the

software. ISA_FRAG represents a data structure that embodies the claimed invention. The ISA_FRAG provides a specific implementation of a payload for transmitting and reassembling IKE fragments.

Exhibit B provides emails from Brian Swander, one of the inventors, that a working implementation of the invention was created and verified to be functional at least as early as December 5, 2001. See email ("I have this implemented, and verified that it work against the NATs in question."). Thus, we had a working ISA_FRAG software solution as early as December 5, 2001. Further, the Description Document also describes the software and provides code with critical elements, such as the fragment_id and fragment_num data. The Description Document also states that the software was coded (i.e., implemented in a software program) at that time. See Description Document, pg. 2, § **Implementation** (This is *currently coded* to be as safe as possible. This means I have traded off optimized buffer management for safety. Also, *I reassemble the entire packet*, and the inject as if it were received from the wire, so there are not new code paths to test." emphasis added). This disclosure demonstrates that we had a robust embodiment of ISA_FRAG-based fragmentation in software on or before December 5, 2001.

c. Several tests verified the working embodiment ISA_FRAG. We completed testing on the prototype software to verify its proper function. See Description Document, pg. 2, § **Testing** ("*I'm done with unit testing*. This includes scale stress testing to make sure there are no leaks." emphasis added). The testing verified function and even some caveats to the software. See Description Document, pg. 2, § **Testing** ("In addition, *I verified that the packet is deleted correctly if all fragments are not properly received, but the next resend is ok.*" emphasis added) We specifically verified proper reassembly as claimed. See Description Document, pg. 2, § **Testing** ("Finally *I made sure that the out of order fragment are correctly reassembled.*" emphasis added). We tested the software against the NATs also. See email ("I have this implemented, *and verified that it works against the NATs inquestion.*" emphasis added). All these tests and comments about the tests were made before submission for a patent application and are prompted by disclosure to team members. Therefore, we tested the ISA_FRAG software to demonstrate that it worked for its intended purpose at least on or before December 5, 2001.

6. We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that statements are made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such false statements may jeopardize the validity of the application or any patent issued thereon.

Date 2/22/07

Brian Swander

Date 2/26/07

Christian Huitema

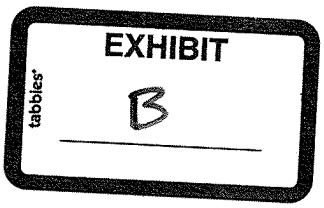


EXHIBIT
B
tabbies

From: Brian Swander
Sent: Wednesday, December 05, 2001 2:26 PM
To: Christian Huitema; Bernard Aboba
Subject: review of IKE fragmentation proposal

I have a proposal for a fix for the IKE fragmentation issue (NATs/Routers dropping IKE frags) that is plaguing the l2tp/ipsec VPN deployment. I have this implemented, and verified that it works against the NATs in question.

Please take a look. This is currently very likely to go into .NET server.

\\networking\pm\net services\ikefrag.doc
<file:///\\networking\pm\net%20services\ikefrag.doc>

bs

From: Brian Swander
Sent: Thursday, December 06, 2001 9:16 AM
To: Christian Huitema; Bernard Aboba
Subject: RE: review of IKE fragmentation proposal

Couple of tradeoffs here. In current scenarios today, if we go to 1500, then we will generally have <4 frags, and be less likely to hit the back-to-back packet drop cases.

The reason I didn't want to jump right to frag is that I believe that this solution should be deprecated as the network infrastructure begins to correctly support IP fragmentation as it is supposed to. Also, leaving the slight delays will give vendors incentive to upgrade their infrastructure.

As for acknowledging the individual frags: the PIC proposal does this by building the knowledge of the frags into the exchange itself, thus modifying the protocol state machine. This is potentially acceptable for a new protocol, but would meet extreme resistance for IKE, an existing protocol.

Also, the number of round trips drastically increases. Of course, we can build in TCP windowing too, but that is getting a little too complicated.

I wanted to define a mechanism to solve fragmentation with absolute minimal intrusion to IKE. That being said, if we determine in the future that we want some sort of ACK or NAK built in, we can always add that. Right now, I'd prefer to keep the reassembly as simple as possible to avoid potential attack scenarios.

bs

-----Original Message-----
From: Christian Huitema
Sent: Thursday, December 06, 2001 8:29 AM
To: Brian Swander; Bernard Aboba

Subject: RE: review of IKE fragmentation proposal

Brian,

Don't optimize the error cases! There is no need to try 1500 first; just go direct for 576. Also, I am concern with the "wait for normal errors" part of your spec. If you know for sure that the other end is a rightly versioned MS implementation, start resending with small frags immediately, and save a cycle of timers. Timers are very irritating when you are waiting for the set-up of a VPN connection.

By the way, do you have any means of performing acknowledgement of fragments? There is another classic failure mode, in which a box is unable to take back to back packets and always drops the 4th or fifth one. Would be nice to be robust against that too.

-- Christian Huitema

-----Original Message-----
From: Brian Swander
Sent: Wednesday, December 05, 2001 2:26 PM
To: Christian Huitema; Bernard Aboba
Subject: review of IKE fragmentation proposal

I have a proposal for a fix for the IKE fragmentation issue (NATs/Routers dropping IKE frags) that is plaguing the l2tp/ipsec VPN deployment. I have this implemented, and verified that it works against the NATs in question.

Please take a look. This is currently very likely to go into .NET server.

\\networking\pm\net services\ikefrag.doc
<file:///\\networking\pm\net%20services\ikefrag.doc>

bs

From: Brian Swander
Sent: Thursday, December 06, 2001 9:47 AM
To: Brian Swander; Christian Huitema; Bernard Aboba
Subject: RE: review of IKE fragmentation proposal

Also, I was mimicking TCPs black hole detection algo as much as possible, since they have implementation experience with that. They start with 1500, then decrease to 576 instead of just sending with the smallest possible MTU. Again, I wanted to minimize frags as much as possible.

bs

-----Original Message-----
From: Brian Swander
Sent: Thursday, December 06, 2001 9:16 AM
To: Christian Huitema; Bernard Aboba
Subject: RE: review of IKE fragmentation proposal

Couple of tradeoffs here. In current scenarios today, if we go to 1500, then we will generally have <4 frags, and be less likely to hit the back-to-back packet drop cases.

The reason I didn't want to jump right to frag is that I believe that this solution should be deprecated as the network infrastructure begins to correctly support IP fragmentation as it is supposed to. Also, leaving the slight delays will give vendors incentive to upgrade their infrastructure.

As for acknowledging the individual frags: the PIC proposal does this by building the knowledge of the frags into the exchange itself, thus modifying the protocol state machine. This is potentially acceptable for a new protocol, but would meet extreme resistance for IKE, an existing protocol.

Also, the number of round trips drastically increases. Of course, we can build in TCP windowing too, but that is getting a little too complicated.

I wanted to define a mechanism to solve fragmentation with absolute minimal intrusion to IKE. That being said, if we determine in the future that we want some sort of ACK or NAK built in, we can always add that. Right now, I'd prefer to keep the reassembly as simple as possible to avoid potential attack scenarios.

bs

-----Original Message-----
From: Christian Huitema
Sent: Thursday, December 06, 2001 8:29 AM
To: Brian Swander; Bernard Aboba
Subject: RE: review of IKE fragmentation proposal

Brian,

Don't optimize the error cases! There is no need to try 1500 first; just go direct for 576. Also, I am concern with the "wait for normal errors" part of your spec. If you know for sure that the other end is a rightly versioned MS implementation, start resending with small frags immediately, and save a cycle of timers. Timers are very irritating when you are waiting for the set-up of a VPN connection.

By the way, do you have any means of performing acknowledgement of fragments? There is another classic failure mode, in which a box is unable to take back to back packets and always drops the 4th or fifth one. Would be nice to be robust against that too.

-- Christian Huitema

-----Original Message-----
From: Brian Swander
Sent: Wednesday, December 05, 2001 2:26 PM
To: Christian Huitema; Bernard Aboba
Subject: review of IKE fragmentation proposal

I have a proposal for a fix for the IKE fragmentation issue (NATs/Routers dropping IKE frags) that is plaguing the l2tp/ipsec VPN deployment. I have this implemented, and verified that it works against the NATs in question.

Please take a look. This is currently very likely to go into .NET server.

\\networking\pm\net services\ikefrag.doc
<file:///\\networking\pm\net%20services\ikefrag.doc>

bs

From: Brian Swander
Sent: Wednesday, January 02, 2002 2:54 PM
To: 'vvolpe@cisco.com'
Cc: William Dixon; Paul Mayfield
Subject: 501 and frag comments

Do you have any comments on this?

Still under NDA:

Apparently, we need to support 501. The current proposal, is to do to immediately float to 501 after detecting the NAT, and move the NAT-D payloads into the first payloads sent by the responder (SA payload). This is minimal effort.

The only issue I have with this approach is that it doesn't allow the peer outside the NAT to be stateless across MM rekeys. In a non-continuous channel MM approach, like we have, it is possible for all MMs to go away, and then for the external peer to need to rekey the MM. That peer will start on 500, which won't be allowed thru the NAT, and the connection will die unless the internal peer decides to rekey. This may be uncommon enough to not cause problems, but it might come back to bite us, too. That is my only concern.

The other option was:

Send an IP UDP ESP "ping" within IKE to detect the UDPESP unfriendly NATs: if it is detected, then either move to 501 to bypass the NAT, or allow the NAT to do the supposed ESP friendliness. This involves timeouts in the UDPESP unfriendly case.

bs

From: Brian Swander
Sent: Thursday, January 03, 2002 8:59 AM
To: 'Victor Volpe'
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com
Subject: RE: 501 and frag comments

Answer to below:

1. Yes, I am speaking of the MTU for the interface. If you subtract the IP/UDP etc, then the MTUs for "IKE" will be smaller.
2. The example is wrong. Yes, we can do without the flags at the expense of the total_frags field in the hdr. However, that is less efficient to encode than a bit flag, and having flags around for extensibility can never hurt.
3. Each frag has a full IKE hdr. Thus, there will be a single reassembly per outstanding SA, which should be adequate even for a gateway, assuming that we are only fragmenting in MM, as is the case today. Of course, you can allow for multiple reassemblies per SA if you really want to. Remember, what you have is more an internal spec, not an internet draft. Thus, there are more internal implementation details than would commonly be in a draft.

4,5. Tradeoffs between complexity of internal state vs. messiness of the detection on the wire. We should make your proposed enhancements optional, since they won't effect interop.

Finally, (still under NDA): Christian is suggesting that we also build in a frag ack (NAK) to say: I didn't get frag #3, please resend it. This can simply be an unauthenticated notify with the data defined to say for which packet ID and fragment num, etc. we need to resend.

Personally, I believe this unnecessary complexity for the near term. He claims that some network infrastructure may consistently drop a particular packet if a stream is being sent. Do you have any experience with this? Any comments?

Thanks

bs

-----Original Message-----

From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Thursday, January 03, 2002 7:40 AM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com
Subject: RE: 501 and frag comments

Hi Brian:

I read through the fragmentation proposal and overall it looks pretty good. The only thing I am not crazy about is the detection mechanism but the alternatives are worse. At least this way, only negotiations with packet loss are affected and not all negotiations. We could (and probably should) provide the user with a knob to set the IKE MTU size if the retries are taking too long or if the user knows ahead of time that there is a fragmentation issue with his NAT device.

Here are specific comments:

1. It looks like your MTU sizes are interface and not IKE based. IKE then needs to account for the MAC and IP headers accordingly so that the IKE MTU values end up being around 1480 and 550.

2. In the FRAG1 and FRAG2 example, you have a total_frags value. total_frags is not in the payload definition, should it be? If it is part of the payload, do you still need the last fragment flag?

3. When you talk about allowing a single reassembly at a time, I am assuming you mean per peer? On the gateway side we have to allow reassembly for multiple peers simultaneously. This requirement should probably be relaxed.

4. It would be nice if the responder could learn the MTU size so that it would not have to go through the same autodetection procedure. This could be just a statement that says "If an ISA_FRAG payload is received, the receiver can assume that the IKE MTU can be no larger than the length of the received IKE packet. The receiver should update its MTU size if it is larger than the learned MTU."

5. For Continuous Channel Mode implementations, I think it would be OK to save the MTU information for rekeys. Again, this would save on the autodetection for every rekey.

I still need to go back and look at the 501 stuff.

Victor

> -----Original Message-----
> From: Brian Swander [mailto:briansw@windows.microsoft.com]
> Sent: Wednesday, January 02, 2002 5:54 PM
> To: Volpe, Victor
> Cc: William Dixon; Paul Mayfield
> Subject: 501 and frag comments
>
> Do you have any comments on this?

Still under NDA:

Apparently, we need to support 501. The current proposal, is to do to immediately float to 501 after detecting the NAT, and move the NAT-D payloads into the first payloads sent by the responder (SA payload). This is minimal effort.

The only issue I have with this approach is that it doesn't allow the peer outside the NAT to be stateless across MM rekeys. In a non-continuous channel MM approach, like we have, it is possible for all MMs to go away, and then for the external peer to need to rekey the MM. That peer will start on 500, which won't be allowed thru the NAT, and the connection will die unless the internal peer decides to rekey. This may be uncommon enough to not cause problems, but it might come back to bite us, too. That is my only concern.

The other option was:

Send an IP UDP ESP "ping" within IKE to detect the UDPESP unfriendly NATs: if it is detected, then either move to 501 to bypass the NAT, or allow the NAT to do the supposed ESP friendliness. This involves timeouts in the UDPESP unfriendly case.

bs

From: Brian Swander
Sent: Thursday, January 03, 2002 9:12 AM
To: 'Victor Volpe'
Cc: William Dixon; Paul Mayfield; danyr@cisco.com
Subject: RE: 501 and frag comments

From my other mail:

Remember, what you have is more an internal spec, not an internet draft. Thus, there are more internal implementation details than would commonly be in a draft.

I tend to agree that 1 second may be too short. For legacy and other reasons, that is what we've got right now. However, there is no reason for you to implement at 1 second if you do not want to. The retransmit intervals, and when you choose to send frags, etc. can be implementation dependent. I can't think of hard reasons to mandate the same algos for determining MTUs and retrans intervals on each side. So long as the frag and reassemble the same.

bs

-----Original Message-----

From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Thursday, January 03, 2002 9:04 AM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com
Subject: RE: 501 and frag comments

Brian:

One additional comment was brought up by Dan Rochefort, our Windows Client Development Mgr. He is concerned that the 1 second retransmission timers can produce some false detections. When this proposal is formalized, the timeouts should probably be longer. If not, they need to be defended with some valid reasons on why they will not cause false detection of a fragmentation problem.

Victor

> -----Original Message-----
> From: Brian Swander [mailto:briansw@windows.microsoft.com]
> Sent: Wednesday, January 02, 2002 5:54 PM
> To: Volpe, Victor
> Cc: William Dixon; Paul Mayfield
> Subject: 501 and frag comments
>
> Do you have any comments on this?
>
> Still under NDA:
>
> Apparently, we need to support 501. The current proposal, is to do to immediately float to 501 after detecting the NAT, and move the NAT-D payloads into the first payloads sent by the responder (SA payload). This is minimal effort.
>
> The only issue I have with this approach is that it doesn't allow the peer outside the NAT to be stateless across MM rekeys. In a non-continuous channel MM approach, like we have, it is possible for all MMs to go away, and then for the external peer to need to rekey the MM. That peer will start on 500, which won't be allowed thru the NAT, and the connection will die unless the internal peer decides to rekey. This may be uncommon enough to not cause problems, but it might come back to bite us, too. That is my only concern.
>
> The other option was:
>
> Send an IP UDP ESP "ping" within IKE to detect the UDPESP unfriendly NATs: if it is detected, then either move to 501 to bypass the NAT, or allow the NAT to do the supposed ESP friendliness. This involves timeouts in the UDPESP unfriendly case.
>
> bs

From: Brian Swander

Sent: Thursday, January 03, 2002 9:26 AM
To: 'Victor Volpe'
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com
Subject: RE: 501 and frag comments

Here's the info for the NAK:

Christian:

Don't bet on the 1/10 case. There is a quite common pattern in which fragment N is lost at every attempt. It happens for example if there is some kind of mistuned leaky bucket component if you are using IP over ATM. Please consider having at least a "NAK" of the form "please resend number N and following". Receiver should send the NAK if it receives an out of sequence packet.

bs

-----Original Message-----
From: Brian Swander
Sent: Wednesday, January 02, 2002 1:37 PM
To: Christian Huitema; Bernard Aboba; David Eitelbach; William Dixon; Ron Cully
Cc: Paul Mayfield
Subject: RE: Questions regarding the ipsec/NAT issue

Building in a NAK (or ACK) mechanism ala TCP doesn't seem essential for this release. Thus, I'd advocate not doing the partial ack for this release. Of course, IKE is extensible enough that this ACK scheme can be built in if necessary later. Indeed, if we write a draft on this, we can define an ACK scheme and make it optional to implement. If deployment experience shows it necessary, we can build it then.

Is this acceptable? I think it would be a very rare case where at lease one "fragment" is dropped on each of the retransmits. If we assume 576 MTU for IKE, a 5000 byte packet is approx 10 "fragments". We retransmit 3 times, and we'd have to have loss rates of 1/10 during each of these sends. Of course, not have an ACK scheme is bad for data transfers, but this should be ok for the IKE control traffic.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Thursday, January 03, 2002 9:15 AM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com
Subject: RE: 501 and frag comments

See below.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Thursday, January 03, 2002 11:59 AM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com
Subject: RE: 501 and frag comments

Answer to below:

1.Yes, I am speaking of the MTU for the interface. If you subtract the IP/UDP etc, then the MTUs for "IKE" will be smaller.

2.The example is wrong. Yes, we can do without the flags at the expense of the total_frags field in the hdr. However, that is less efficient to encode than a bit flag, and having flags around for extensibility can never hurt.

(VV) - I agree that the flags is a better way of handling this.

1.Each frag has a full IKE hdr. Thus, there will be a single reassembly per outstanding SA, which should be adequate even for a gateway, assuming that we are only fragmenting in MM, as is the case today. Of course, you can allow for multiple reassemblies per SA if you really want to. Remember, what you have is more an internal spec, not an internet draft. Thus, there are more internal implementation details than would commonly be in a draft.

(VV) - OK, that's what I thought but wanted to check.

4,5. Tradeoffs between complexity of internal state vs. messiness of the detection on the wire. We should make your proposed enhancements optional, since they won't effect interop.

(VV) - Sounds good.

Finally, (still under NDA): Christian is suggesting that we also build in a frag ack (NAK) to say: I didn't get frag #3, please resend it. This can simply be an unauthenticated notify with the data defined to say for which packet ID and fragment num, etc. we need to resend.

Personally, I believe this unnecessary complexity for the near term. He claims that some network infrastructure may consistently drop a particular packet if a stream is being sent. Do you have any experience with this? Any comments?

(VV) - I do not have any experience with this but am not sure that the ACK would solve the problem anyway. Is he saying that #3 would always get dropped if it is sent with the other fragments but would not get dropped if is sent alone. This does not sound right to me but I guess I do not know. It would be nice to not have to implement the notify. I will forward this to some people here to see if they have seen this type of a problem.

Thanks

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Thursday, January 03, 2002 7:40 AM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com
Subject: RE: 501 and frag comments

Hi Brian:

I read through the fragmentation proposal and overall it looks pretty good. The only thing I am not crazy about is the detection mechanism but the alternatives are worse. At least this way, only negotiations with packet loss are affected and not all negotiations. We could (and probably should) provide the user with a knob to set the IKE MTU size if the retries are taking too long or if the user knows ahead of time that there is a fragmentation issue with his NAT device.

Here are specific comments:

1. It looks like your MTU sizes are interface and not IKE based. IKE then needs to account for the MAC and IP headers accordingly so that the IKE MTU values end up being around 1480 and 550.

2. In the FRAG1 and FRAG2 example, you have a total_frags value. total_frags is not in the payload definition, should it be? If it is part of the payload, do you still need the last fragment flag?

3. When you talk about allowing a single reassembly at a time, I am assuming you mean per peer? On the gateway side we have to allow reassembly for multiple peers simultaneously. This requirement should probably be relaxed.

4. It would be nice if the responder could learn the MTU size so that it would not have to go through the same autodetection procedure. This could be just a statement that says "If an ISA_FRAG payload is received, the receiver can assume that the IKE MTU can be no larger than the length of the received IKE packet. The receiver should update its MTU size if it is larger than the learned MTU."

5. For Continuous Channel Mode implementations, I think it would be OK to save the MTU information for rekeys. Again, this would save on the autodetection for every rekey.

I still need to go back and look at the 501 stuff.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Wednesday, January 02, 2002 5:54 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield
Subject: 501 and frag comments

Do you have any comments on this?

Still under NDA:

Apparently, we need to support 501. The current proposal, is to do to immediately float to 501 after detecting the NAT, and move the NAT-D payloads into the first payloads sent by the responder (SA payload). This is minimal effort.

The only issue I have with this approach is that it doesn't allow the peer outside the NAT to be stateless across MM rekeys. In a non-continuous channel MM approach, like we have, it is possible for all MMs to go away, and then for the external peer to need to rekey the MM. That peer will start on 500, which won't be allowed thru the NAT, and the connection will die unless the internal peer decides to rekey. This may be uncommon enough to not cause problems, but it might come back to bite us, too. That is my only concern.

The other option was:

Send an IP UDP ESP "ping" within IKE to detect the UDPESP unfriendly NATs: if it is detected, then either move to 501 to bypass the NAT, or allow the NAT to do the supposed ESP friendliness. This involves timeouts in the UDPESP unfriendly case.

bs

From: Brian Swander
Sent: Thursday, January 03, 2002 3:12 PM
To: 'Victor Volpe'
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com; izubenko@cisco.com
Subject: RE: 501 and frag comments

No, we don't need a containing payload, since we fragment the whole IKE packet, including the IKE hdr. So, the full blow up is:

Original non-fragmented packet: IKEHDR (Encrypt, nextp=ID) Data

becomes:

IKEHDR (Noencrypt, nextp=ISAFrag) IKEHDR(Encrypt,nextp=ID), beginning of data

IKEHDR (Noencrypt, nextp=ISAFrag) more data

Etc.

Thus, instead of defining the extra fields that we care about, (nextp and hdr flags at least) in the ISA_FRAG header, I thought it much simpler to just include the original hdr, too.

This solves your concern, doesn't it? Also, since the ISA_FRAG header isn't protected, you'd have to set these fields in every frag, validate they were the same each time, etc. This duplication just didn't seem worth it.

I don't see why you'd want to change to all FFs. I know all 0's in an invalid cookie in some RFC, and all FFs is probably valid, and I don't see what moving to all FFs would solve.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Thursday, January 03, 2002 1:59 PM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com; izubenko@cisco.com
Subject: RE: 501 and frag comments

Hi Brian:

One of our developers (Igor Zubenko on the CC list) has been implementing the fragmentation proposal and got it working but he ran into a few issues. One of the issues, the use of the total_frag value, we have already talked about. The other issue has to do with the ability to make the frag payload more generic. Since the contained payload value is overwritten in the IKE header, this will only work for MM pkts 5 and 6 (or for a single known first payload). Can you add a "contained_payload" field or something like that to the frag payload. This will remove the limitation and allow us to use it in different areas. For example, we are concerned that some transaction mode pkts are getting close to the MTU size.

This will give us a way to get around that.

Also, Igor asked about the non-IKE marker in the port 500 version of NAT-T. His question is: why not use a cookie of all FFs. I know this was talked about early on but I do not remember the reasons why it was dismissed. Was it that all FFs is a valid cookie. If that was the only issue, NAT-T could specify that an all FF cookie has "special" meaning. Of course, there would still be the problem of IPsec aware NATs not changing the source port on IKE pkts.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Thursday, January 03, 2002 12:26 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com
Subject: RE: 501 and frag comments

Here's the info for the NAK:

Christian:

Don't bet on the 1/10 case. There is a quite common pattern in which fragment N is lost at every attempt. It happens for example if there is some kind of mistuned leaky bucket component if you are using IP over ATM. Please consider having at least a "NAK" of the form "please resend number N and following". Receiver should send the NAK if it receives an out of sequence packet.

bs

-----Original Message-----
From: Brian Swander
Sent: Wednesday, January 02, 2002 1:37 PM
To: Christian Huitema; Bernard Aboba; David Eitelbach; William Dixon; Ron Cully
Cc: Paul Mayfield
Subject: RE: Questions regarding the ipsec/NAT issue

Building in a NAK (or ACK) mechanism ala TCP doesn't seem essential for this release. Thus, I'd advocate not doing the partial ack for this release. Of course, IKE is extensible enough that this ACK scheme can be built in if necessary later. Indeed, if we write a draft on this, we can define an ACK scheme and make it optional to implement. If deployment experience shows it necessary, we can build it then.

Is this acceptable? I think it would be a very rare case where at lease one "fragment" is dropped on each of the retransmits. If we assume 576 MTU for IKE, a 5000 byte packet is approx 10 "fragments". We retransmit 3 times, and we'd have to have loss rates of 1/10 during each of these sends. Of course, not have an ACK

scheme is bad for data transfers, but this should be ok for the IKE control traffic.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Thursday, January 03, 2002 9:15 AM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com
Subject: RE: 501 and frag comments

See below.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Thursday, January 03, 2002 11:59 AM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com
Subject: RE: 501 and frag comments

Answer to below:

1.Yes, I am speaking of the MTU for the interface. If you subtract the IP/UDP etc, then the MTUs for "IKE" will be smaller.

2.The example is wrong. Yes, we can do without the flags at the expense of the total_frags field in the hdr. However, that is less efficient to encode than a bit flag, and having flags around for extensibility can never hurt.

(VV) - I agree that the flags is a better way of handling this.

1.Each frag has a full IKE hdr. Thus, there will be a single reassembly per outstanding SA, which should be adequate even for a gateway, assuming that we are only fragmenting in MM, as is the case today. Of course, you can allow for multiple reassemblies per SA if you really want to. Remember, what you have is more an internal spec, not an internet draft. Thus, there are more internal implementation details than would commonly be in a draft.

(VV) - OK, that's what I thought but wanted to check.

4,5. Tradeoffs between complexity of internal state vs. messiness of the detection on the wire. We should make your proposed enhancements optional,

since they won't effect interop.

(VV) - Sounds good.

Finally, (still under NDA): Christian is suggesting that we also build in a frag ack (NAK) to say: I didn't get frag #3, please resend it. This can simply be an unauthenticated notify with the data defined to say for which packet ID and fragment num, etc. we need to resend.

Personally, I believe this unnecessary complexity for the near term. He claims that some network infrastructure may consistently drop a particular packet if a stream is being sent. Do you have any experience with this? Any comments?

(VV) - I do not have any experience with this but am not sure that the ACK would solve the problem anyway. Is he saying that #3 would always get dropped if it is sent with the other fragments but would not get dropped if is sent alone. This does not sound right to me but I guess I do not know. It would be nice to not have to implement the notify. I will forward this to some people here to see if they have seen this type of a problem.

Thanks

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Thursday, January 03, 2002 7:40 AM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com
Subject: RE: 501 and frag comments

Hi Brian:

I read through the fragmentation proposal and overall it looks pretty good. The only thing I am not crazy about is the detection mechanism but the alternatives are worse. At least this way, only negotiations with packet loss are affected and not all negotiations. We could (and probably should) provide the user with a knob to set the IKE MTU size if the retries are taking too long or if the user knows ahead of time that there is a fragmentation issue with his NAT device.

Here are specific comments:

1. It looks like your MTU sizes are interface and not IKE based. IKE then needs to account for the MAC and IP headers accordingly so that the IKE MTU values end up being around 1480 and 550.

2. In the FRAG1 and FRAG2 example, you have a total_frags value. total_frags is not in the payload definition, should it be? If it is part of the payload, do you still need the last fragment flag?

3. When you talk about allowing a single reassembly at a time, I am assuming you mean per peer? On the gateway side we have to allow reassembly for multiple peers simultaneously. This requirement should probably be relaxed.

4. It would be nice if the responder could learn the MTU size so that it would not have to go through the same autodetection procedure. This could be just a statement that says "If an ISA_FRAG payload is received, the receiver can assume that the IKE MTU can be no larger than the length of the received IKE packet. The receiver should update its MTU size if it is larger than the learned MTU."

5. For Continuous Channel Mode implementations, I think it would be OK to save the MTU information for rekeys. Again, this would save on the autodetection for every rekey.

I still need to go back and look at the 501 stuff.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Wednesday, January 02, 2002 5:54 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield
Subject: 501 and frag comments

Do you have any comments on this?

Still under NDA:

Apparently, we need to support 501. The current proposal, is to do to immediately float to 501 after detecting the NAT, and move the NAT-D payloads into the first payloads sent by the responder (SA payload). This is minimal effort.

The only issue I have with this approach is that it doesn't allow the peer outside the NAT to be stateless across MM rekeys. In a non-continuous channel MM approach, like we have, it is possible for all MMs to go away, and then for the external peer to need to rekey the MM. That peer will start on 500, which won't be allowed thru the NAT, and the connection will die unless the internal peer decides to rekey. This may be uncommon enough to not cause problems, but it might come back to bite us, too. That is my only concern.

The other option was:

Send an IP UDP ESP "ping" within IKE to detect the UDPESP unfriendly NATs: if it is detected, then either move to 501 to bypass the NAT, or allow the NAT to do the supposed ESP friendliness. This involves timeouts in the UDPESP unfriendly case.

bs

From: Brian Swander
Sent: Thursday, January 03, 2002 4:49 PM
To: Kartik Murthy
Subject: FW: 501 and frag comments

I'll get you on the thread proper as soon as I send another message.

bs

-----Original Message-----
From: Brian Swander
Sent: Thursday, January 03, 2002 3:12 PM
To: 'Victor Volpe'
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com; izubenko@cisco.com
Subject: RE: 501 and frag comments

No, we don't need a containing payload, since we fragment the whole IKE packet, including the IKE hdr. So, the full blow up is:

Original non-fragmented packet: IKEHDR (Encrypt, nextp=ID) Data

becomes:

IKEHDR (Noencrypt, nextp=ISAFrag) IKEHDR(Encrypt,nextp=ID), beginning of data

IKEHDR (Noencrypt, nextp=ISAFrag) more data

Etc.

Thus, instead of defining the extra fields that we care about, (nextp and hdr flags at least) in the ISA_FRAG header, I thought it much simpler to just include the original hdr, too.

This solves your concern, doesn't it? Also, since the ISA_FRAG header isn't protected, you'd have to set these fields in every frag, validate they were the same each time, etc. This duplication just didn't seem worth it.

I don't see why you'd want to change to all FFs. I know all 0's in an invalid cookie in some RFC, and all FFs is probably valid, and I don't see what moving to all FFs would solve.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Thursday, January 03, 2002 1:59 PM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com; izubenko@cisco.com
Subject: RE: 501 and frag comments

Hi Brian:

One of our developers (Igor Zubenko on the CC list) has been implementing the fragmentation proposal and got it working but he ran into a few issues. One of the issues, the use of the total_frag value, we have already talked about. The other issue has to do with the ability to make the frag payload more generic. Since the contained payload value is overwritten in the IKE header, this will only work for MM pkts 5 and 6 (or for a single known first payload). Can you add a "contained_payload" field or something like that to the frag payload. This will remove the limitation and allow us to use it in different areas. For example, we are concerned that some transaction mode pkts are getting close to the MTU size. This will give us a way to get around that.

Also, Igor asked about the non-IKE marker in the port 500 version of NAT-T. His question is: why not use a cookie of all FFs. I know this was talked about early on but I do not remember the reasons why it was dismissed. Was it that all FFs is a valid cookie. If that was the only issue, NAT-T could specify that an all FF cookie has "special" meaning. Of course, there would still be the problem of IPsec aware NATs not changing the source port on IKE pkts.

Victor

> -----Original Message-----
> From: Brian Swander [mailto:briansw@windows.microsoft.com]
> Sent: Thursday, January 03, 2002 12:26 PM
> To: Volpe, Victor
> Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com
> Subject: RE: 501 and frag comments
>
> Here's the info for the NAK:
>
> Christian:
>
> Don't bet on the 1/10 case. There is a quite common pattern in which

fragment N is lost at every attempt. It happens for example if there is some kind of mistuned leaky bucket component if you are using IP over ATM. Please consider having at least a "NAK" of the form "please resend number N and following". Receiver should send the NAK if it receives an out of sequence packet.

bs

-----Original Message-----
From: Brian Swander
Sent: Wednesday, January 02, 2002 1:37 PM
To: Christian Huitema; Bernard Aboba; David Eitelbach; William Dixon; Ron Cully
Cc: Paul Mayfield
Subject: RE: Questions regarding the ipsec/NAT issue

Building in a NAK (or ACK) mechanism ala TCP doesn't seem essential for this release. Thus, I'd advocate not doing the partial ack for this release. Of course, IKE is extensible enough that this ACK scheme can be built in if necessary later. Indeed, if we write a draft on this, we can define an ACK scheme and make it optional to implement. If deployment experience shows it necessary, we can build it then.

Is this acceptable? I think it would be a very rare case where at lease one "fragment" is dropped on each of the retransmits. If we assume 576 MTU for IKE, a 5000 byte packet is approx 10 "fragments". We retransmit 3 times, and we'd have to have loss rates of 1/10 during each of these sends. Of course, not have an ACK scheme is bad for data transfers, but this should be ok for the IKE control traffic.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Thursday, January 03, 2002 9:15 AM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com
Subject: RE: 501 and frag comments

See below.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Thursday, January 03, 2002 11:59 AM

To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com
Subject: RE: 501 and frag comments

Answer to below:

1.Yes, I am speaking of the MTU for the interface. If you subtract the IP/UDP etc, then the MTUs for "IKE" will be smaller.

2.The example is wrong. Yes, we can do without the flags at the expense of the total_frags field in the hdr. However, that is less efficient to encode than a bit flag, and having flags around for extensibility can never hurt.

(VV) - I agree that the flags is a better way of handling this.

1.Each frag has a full IKE hdr. Thus, there will be a single reassembly per outstanding SA, which should be adequate even for a gateway, assuming that we are only fragmenting in MM, as is the case today. Of course, you can allow for multiple reassemblies per SA if you really want to. Remember, what you have is more an internal spec, not an internet draft. Thus, there are more internal implementation details than would commonly be in a draft.

(VV) - OK, that's what I thought but wanted to check.

4,5. Tradeoffs between complexity of internal state vs. messiness of the detection on the wire. We should make your proposed enhancements optional, since they won't effect interop.

(VV) - Sounds good.

Finally, (still under NDA): Christian is suggesting that we also build in a frag ack (NAK) to say: I didn't get frag #3, please resend it. This can simply be an unauthenticated notify with the data defined to say for which packet ID and fragment num, etc. we need to resend.

Personally, I believe this unnecessary complexity for the near term. He claims that some network infrastructure may consistently drop a particular packet if a stream is being sent. Do you have any experience with this? Any comments?

(VV) - I do not have any experience with this but am not sure that the ACK would solve the problem anyway. Is he saying that #3 would always get dropped if it is sent with the other fragments but would not get dropped if is sent alone. This does not sound right to me but I guess I do not know. It would be nice to not have to implement the notify. I will forward this to some people here to see if they have seen this type of a problem.

Thanks

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Thursday, January 03, 2002 7:40 AM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com
Subject: RE: 501 and frag comments

Hi Brian:

I read through the fragmentation proposal and overall it looks pretty good. The only thing I am not crazy about is the detection mechanism but the alternatives are worse. At least this way, only negotiations with packet loss are affected and not all negotiations. We could (and probably should) provide the user with a knob to set the IKE MTU size if the retries are taking too long or if the user knows ahead of time that there is a fragmentation issue with his NAT device.

Here are specific comments:

1. It looks like your MTU sizes are interface and not IKE based. IKE then needs to account for the MAC and IP headers accordingly so that the IKE MTU values end up being around 1480 and 550.

2. In the FRAG1 and FRAG2 example, you have a total_frags value. total_frags is not in the payload definition, should it be? If it is part of the payload, do you still need the last fragment flag?

3. When you talk about allowing a single reassembly at a time, I am assuming you mean per peer? On the gateway side we have to allow reassembly for multiple peers simultaneously. This requirement should probably be relaxed.

4. It would be nice if the responder could learn the MTU size so that it would not have to go through the same autodetection procedure. This could be just a statement that says "If an ISA_FRAG payload is received, the receiver can assume that the IKE MTU can be no larger than the length of the received IKE packet. The receiver should update its MTU size if it is larger than the learned MTU."

5. For Continuous Channel Mode implementations, I think it would be OK to save the MTU information for rekeys. Again, this would save on the autodetection for every rekey.

I still need to go back and look at the 501 stuff.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Wednesday, January 02, 2002 5:54 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield
Subject: 501 and frag comments

Do you have any comments on this?

Still under NDA:

Apparently, we need to support 501. The current proposal, is to do to immediately float to 501 after detecting the NAT, and move the NAT-D payloads into the first payloads sent by the responder (SA payload). This is minimal effort.

The only issue I have with this approach is that it doesn't allow the peer outside the NAT to be stateless across MM rekeys. In a non-continuous channel MM approach, like we have, it is possible for all MMs to go away, and then for the external peer to need to rekey the MM. That peer will start on 500, which won't be allowed thru the NAT, and the connection will die unless the internal peer decides to rekey. This may be uncommon enough to not cause problems, but it might come back to bite us, too. That is my only concern.

The other option was:

Send an IP UDP ESP "ping" within IKE to detect the UDPESP unfriendly NATs: if it is detected, then either move to 501 to bypass the NAT, or allow the NAT to do the supposed ESP friendliness. This involves timeouts in the UDPESP unfriendly case.

bs

From: Brian Swander
Sent: Friday, January 04, 2002 8:52 AM
To: 'Dany Rochefort'; vvolpe@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com; Kartik Murthy
Subject: RE: 501 and frag comments

This is extra complexity for little gain, IMHO. If we are fragmenting an encrypted payload (as is the common case for the ID payload), then all that can be modified is frag header fields. If they modify the encrypted data within the frag, that is identical to modifying an unfragmented, encrypted ID payload, which IKE is already robust against.

The worst this can do is cause the packet to fail reassembly and be lost, or scramble the packet during reassembly. Any attacker that can modify bits on the wire can already force IKE packets to be dropped anyway. Also, an attacker can similarily scramble the packet, encrypted or otherwise, and hashing/validation of

the whole packet solves this (which we have once crypto is active).

Thus, I don't see how protecting the frag header is worth the effort. Also, then we'll have different semantics for before and after crypto keys are generated, and the problem gets much much tougher.

bs

-----Original Message-----
From: Dany Rochefort [mailto:danyr@cisco.com]
Sent: Friday, January 04, 2002 6:44 AM
To: Brian Swander; vvolpe@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com
Subject: RE: 501 and frag comments

Hi Brian,

Regarding the IKE_FRAG payload. In your proposal, you mention that someone could modify some of the data since it's not encrypted. I was wondering if you had considered HASHING the IKE_FRAG itself to allow the peer to confirm that the IKE_FRAG is intact? I realize this does create some additional overhead.

-dany

> -----Original Message-----
> From: Brian Swander [mailto:briansw@windows.microsoft.com]
> Sent: Thursday, January 03, 2002 6:12 PM
> To: Volpe, Victor
> Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com; Zubenko, Igor
> Subject: RE: 501 and frag comments
>
> No, we don't need a containing payload, since we fragment the whole IKE packet, including the IKE hdr. So, the full blow up is:
>
> Original non-fragmented packet: IKEHDR (Encrypt, nextp=ID) Data
>
> becomes:

IKEHDR (Noencrypt, nextp=ISAFrag) IKEHDR(Encrypt,nextp=ID), beginning of data

IKEHDR (Noencrypt, nextp=ISAFrag) more data

Etc.

Thus, instead of defining the extra fields that we care about, (nextp and hdr flags at least) in the ISA_FRAG header, I thought it much simpler to just include the original hdr, too.

This solves your concern, doesn't it? Also, since the ISA_FRAG header isn't protected, you'd have to set these fields in every frag, validate they were the same each time, etc. This duplication just didn't seem worth it.

I don't see why you'd want to change to all FFs. I know all 0's in an invalid cookie in some RFC, and all FFs is probably valid, and I don't see what moving to all FFs would solve.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Thursday, January 03, 2002 1:59 PM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com; izubenko@cisco.com
Subject: RE: 501 and frag comments

Hi Brian:

One of our developers (Igor Zubenko on the CC list) has been implementing the fragmentation proposal and got it working but he ran into a few issues. One of the issues, the use of the total_frag value, we have already talked about. The other issue has to do with the ability to make the frag payload more generic. Since the contained payload value is overwritten in the IKE header, this will only work for MM pkts 5 and 6 (or for a single known first payload). Can you add a "contained_payload" field or something like that to the frag payload. This will remove the limitation and allow us to use it in different areas. For example, we are concerned that some transaction mode pkts are getting close to the MTU size. This will give us a way to get around that.

Also, Igor asked about the non-IKE marker in the port 500 version of NAT-T. His question is: why not use a cookie of all FFs. I know this was talked about early on but I do not remember the reasons why it was dismissed. Was it that all FFs is a valid cookie. If that was the only issue, NAT-T could specify that an all FF cookie has "special" meaning. Of course, there would still be the problem of

IPsec aware NATs not changing the source port on IKE pkts.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Thursday, January 03, 2002 12:26 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com
Subject: RE: 501 and frag comments

Here's the info for the NAK:

Christian:

Don't bet on the 1/10 case. There is a quite common pattern in which fragment N is lost at every attempt. It happens for example if there is some kind of mistuned leaky bucket component if you are using IP over ATM. Please consider having at least a "NAK" of the form "please resend number N and following". Receiver should send the NAK if it receives an out of sequence packet.

bs

-----Original Message-----
From: Brian Swander
Sent: Wednesday, January 02, 2002 1:37 PM
To: Christian Huitema; Bernard Aboba; David Eitelbach; William Dixon; Ron Cully
Cc: Paul Mayfield
Subject: RE: Questions regarding the ipsec/NAT issue

Building in a NAK (or ACK) mechanism ala TCP doesn't seem essential for this release. Thus, I'd advocate not doing the partial ack for this release. Of course, IKE is extensible enough that this ACK scheme can be built in if necessary later. Indeed, if we write a draft on this, we can define an ACK scheme and make it optional to implement. If deployment experience shows it necessary, we can build it then.

Is this acceptable? I think it would be a very rare case where at lease one "fragment" is dropped on each of the retransmits. If we assume 576 MTU for IKE, a 5000 byte packet is approx 10 "fragments". We retransmit 3 times, and we'd have to have loss rates of 1/10 during each of these sends. Of course, not have an ACK scheme is bad for data transfers, but this should be ok for the IKE control traffic.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Thursday, January 03, 2002 9:15 AM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com
Subject: RE: 501 and frag comments

See below.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Thursday, January 03, 2002 11:59 AM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com
Subject: RE: 501 and frag comments

Answer to below:

1.Yes, I am speaking of the MTU for the interface. If you subtract the IP/UDP etc, then the MTUs for "IKE" will be smaller.

2.The example is wrong. Yes, we can do without the flags at the expense of the total_frags field in the hdr. However, that is less efficient to encode than a bit flag, and having flags around for extensibility can never hurt.

(VV) - I agree that the flags is a better way of handling this.

1.Each frag has a full IKE hdr. Thus, there will be a single reassembly per outstanding SA, which should be adequate even for a gateway, assuming that we are only fragmenting in MM, as is the case today. Of course, you can allow for multiple reassemblies per SA if you really want to. Remember, what you have is more an internal spec, not an internet draft. Thus, there are more internal implementation details than would commonly be in a draft.

(VV) - OK, that's what I thought but wanted to check.

4,5. Tradeoffs between complexity of internal state vs. messiness of the detection on the wire. We should make your proposed enhancements optional, since they won't effect interop.

(VV) - Sounds good.

Finally, (still under NDA): Christian is suggesting that we also build in a frag ack (NAK) to say: I didn't get frag #3, please resend it. This can simply be an unauthenticated notify with the data defined to say for which packet ID and fragment num, etc. we need to resend.

Personally, I believe this unnecessary complexity for the near term. He claims that some network infrastructure may consistently drop a particular packet if a stream is being sent. Do you have any experience with this? Any comments?

(VV) - I do not have any experience with this but am not sure that the ACK would solve the problem anyway. Is he saying that #3 would always get dropped if it is sent with the other fragments but would not get dropped if is sent alone. This does not sound right to me but I guess I do not know. It would be nice to not have to implement the notify. I will forward this to some people here to see if they have seen this type of a problem.

Thanks

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Thursday, January 03, 2002 7:40 AM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com
Subject: RE: 501 and frag comments

Hi Brian:

I read through the fragmentation proposal and overall it looks pretty good. The only thing I am not crazy about is the detection mechanism but the alternatives are worse. At least this way, only negotiations with packet loss are affected and not all negotiations. We could (and probably should) provide the user with a knob to set the IKE MTU size if the retries are taking too long or if the user knows ahead of time that there is a fragmentation issue with his NAT device.

Here are specific comments:

1. It looks like your MTU sizes are interface and not IKE based. IKE then needs to account for the MAC and IP headers accordingly so that the IKE MTU values end up being around 1480 and 550.

2. In the FRAG1 and FRAG2 example, you have a total_frags value. total_frags is not in the payload definition, should it be? If it is part of the payload, do you still need the last fragment flag?

3. When you talk about allowing a single reassembly at a time, I am assuming you mean per peer? On the gateway side we have to allow reassembly for multiple peers simultaneously. This requirement should probably be relaxed.

4. It would be nice if the responder could learn the MTU size so that it would not have to go through the same autodetection procedure. This could be just a statement that says "If an ISA_FRAG payload is received, the receiver can assume that the IKE MTU can be no larger than the length of the received IKE packet. The receiver should update its MTU size if it is larger than the learned MTU."

5. For Continuous Channel Mode implementations, I think it would be OK to save the MTU information for rekeys. Again, this would save on the autodetection for every rekey.

I still need to go back and look at the 501 stuff.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Wednesday, January 02, 2002 5:54 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield
Subject: 501 and frag comments

Do you have any comments on this?

Still under NDA:

Apparently, we need to support 501. The current proposal, is to do to immediately float to 501 after detecting the NAT, and move the NAT-D payloads into the first payloads sent by the responder (SA payload). This is minimal effort.

The only issue I have with this approach is that it doesn't allow the peer outside the NAT to be stateless across MM rekeys. In a non-continuous channel MM approach, like we have, it is possible for all MMs to go away, and then for the external peer to need to rekey the MM. That peer will start on 500, which won't be allowed thru the NAT, and the connection will die unless the internal peer decides to rekey. This may be uncommon enough to not cause problems, but it might come back to bite us, too. That is my only concern.

The other option was:

Send an IP UDP ESP "ping" within IKE to detect the

UDPESP unfriendly NATs: if it is detected, then either move to 501 to bypass the NAT, or allow the NAT to do the supposed ESP friendliness. This involves timeouts in the UDPESP unfriendly case.

bs

From: Brian Swander
Sent: Friday, January 04, 2002 9:54 AM
To: 'Dany Rochefort'; vvolpe@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com; Kartik Murthy
Subject: RE: 501 and frag comments

Are we agreed to this design? Any other outstanding issues?

bs

-----Original Message-----
From: Dany Rochefort [mailto:danyr@cisco.com]
Sent: Friday, January 04, 2002 9:22 AM
To: Brian Swander; vvolpe@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com; Kartik Murthy
Subject: RE: 501 and frag comments

Agreed. I was merely trying to protect the reassembly logic, but you bring up very good points that would minimize its usefulness.

-dany

> -----Original Message-----
> From: Brian Swander [mailto:briansw@windows.microsoft.com]
> Sent: Friday, January 04, 2002 11:52 AM
> To: Rochefort, Dany; Volpe, Victor
> Cc: William Dixon; Paul Mayfield; psd@cisco.com; Zubenko, Igor; Kartik Murthy
> Subject: RE: 501 and frag comments
>
> This is extra complexity for little gain, IMHO. If we are fragmenting an encrypted payload (as is the common case for the ID payload), then all that can be modified is frag header fields. If they modify the encrypted data within the frag, that is identical to modifying an unfragmented, encrypted ID payload, which IKE is already robust against.
>
> The worst this can do is cause the packet to fail reassembly and be lost, or scramble the packet during reassembly. Any attacker that can modify bits on the wire can already force IKE packets to be dropped anyway. Also, an attacker can similarily scramble the packet, encrypted or otherwise, and hashing/validation of the whole packet solves this (which we have once crypto is active).

Thus, I don't see how protecting the frag header is worth the effort. Also, then we'll have different semantics for before and after crypto keys are generated, and the problem gets much much tougher.

bs

-----Original Message-----
From: Dany Rochefort [mailto:danyr@cisco.com]
Sent: Friday, January 04, 2002 6:44 AM
To: Brian Swander; vvolpe@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com
Subject: RE: 501 and frag comments

Hi Brian,

Regarding the IKE_FRAG payload. In your proposal, you mention that someone could modify some of the data since it's not encrypted. I was wondering if you had considered HASHING the IKE_FRAG itself to allow the peer to confirm that the IKE_FRAG is intact? I realize this does create some additional overhead.

-dany

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Thursday, January 03, 2002 6:12 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com; Zubenko, Igor
Subject: RE: 501 and frag comments

No, we don't need a containing payload, since we fragment the whole IKE packet, including the IKE hdr. So, the full blow up is:

Original non-fragmented packet: IKEHDR (Encrypt, nextp=ID) Data

becomes:

IKEHDR (Noencrypt, nextp=ISAFrag) IKEHDR(Encrypt,nextp=ID), beginning of data

IKEHDR (Noencrypt, nextp=ISAFrag) more data

Etc.

Thus, instead of defining the extra fields that we care about, (nextp and hdr flags at least) in the ISA_FRAG header, I thought it much simpler to just include the original hdr, too.

This solves your concern, doesn't it? Also, since the ISA_FRAG header isn't protected, you'd have to set these fields in every frag, validate they were the same each time, etc. This duplication just didn't seem worth it.

I don't see why you'd want to change to all FFs. I know all 0's in an invalid cookie in some RFC, and all FFs is probably valid, and I don't see what moving to all FFs would solve.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Thursday, January 03, 2002 1:59 PM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com; izubenko@cisco.com
Subject: RE: 501 and frag comments

Hi Brian:

One of our developers (Igor Zubenko on the CC list) has been implementing the fragmentation proposal and got it working but he ran into a few issues. One of the issues, the use of the total_frag value, we have already talked about. The other issue has to do with the ability to make the frag payload more generic. Since the contained payload value is overwritten in the IKE header, this will only work for MM pkts 5 and 6 (or for a single known first payload). Can you add a "contained_payload" field or something like that to the frag payload. This will remove the limitation and allow us to use it in different areas. For example, we are concerned that some transaction mode pkts are getting close to the MTU size. This will give us a way to get around that.

Also, Igor asked about the non-IKE marker in the port 500 version of NAT-T. His question is: why not use a cookie of all FFs. I know this was talked about early on but I do not remember the reasons why it was dismissed. Was it that all FFs is a valid cookie. If that was the only issue, NAT-T could specify that an all FF cookie has "special" meaning. Of course, there would still be the problem of IPsec aware NATs not changing the source port on IKE pkts.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Thursday, January 03, 2002 12:26 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com
Subject: RE: 501 and frag comments

Here's the info for the NAK:

Christian:

Don't bet on the 1/10 case. There is a quite common pattern in which fragment N is lost at every attempt. It happens for example if there is some kind of mistuned leaky bucket component if you are using IP over ATM. Please consider having at least a "NAK" of the form "please resend number N and following". Receiver should send the NAK if it receives an out of sequence packet.

bs

-----Original Message-----
From: Brian Swander
Sent: Wednesday, January 02, 2002 1:37 PM
To: Christian Huitema; Bernard Aboba; David Eitelbach; William Dixon; Ron Cully
Cc: Paul Mayfield
Subject: RE: Questions regarding the ipsec/NAT issue

Building in a NAK (or ACK) mechanism ala TCP doesn't seem essential for this release. Thus, I'd advocate not doing the partial ack for this release. Of course, IKE is extensible enough that this ACK scheme can be built in if necessary later. Indeed, if we write a draft on this, we can define an ACK scheme and make it optional to implement. If deployment experience shows it necessary, we can build it then.

Is this acceptable? I think it would be a very rare case where at lease one "fragment" is dropped on each of the retransmits. If we assume 576 MTU for IKE, a 5000 byte packet is approx 10 "fragments". We retransmit 3 times, and we'd have to have loss rates of 1/10 during each of these sends. Of course, not have an ACK scheme is bad for data transfers, but this should be ok for the IKE control traffic.

bs

-----Original Message-----

From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Thursday, January 03, 2002 9:15 AM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com;
psd@cisco.com
Subject: RE: 501 and frag comments

See below.

Victor

-----Original Message-----
From: Brian Swander
[mailto:briansw@windows.microsoft.com]
Sent: Thursday, January 03, 2002 11:59 AM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany;
psd@cisco.com
Subject: RE: 501 and frag comments

Answer to below:

1.Yes, I am speaking of the MTU for the interface. If you subtract the IP/UDP etc, then the MTUs for "IKE" will be smaller.

2.The example is wrong. Yes, we can do without the flags at the expense of the total_frags field in the hdr. However, that is less efficient to encode than a bit flag, and having flags around for extensibility can never hurt.

(VV) - I agree that the flags is a better way of handling this.

1.Each frag has a full IKE hdr. Thus, there will be a single reassembly per outstanding SA, which should be adequate even for a gateway, assuming that we are only fragmenting in MM, as is the case today. Of course, you can allow for multiple reassemblies per SA if you really want to. Remember, what you have is more an internal spec, not an internet draft. Thus, there are more internal implementation details than would commonly be in a draft.

(VV) - OK, that's what I thought but wanted to check.

4,5. Tradeoffs between complexity of internal state vs. messiness of the detection on the wire. We should make your proposed enhancements optional, since they won't effect interop.

(VV) - Sounds good.

Finally, (still under NDA): Christian is suggesting that we also build in a frag ack (NAK) to say: I didn't get frag #3, please resend it. This can simply be an unauthenticated notify with the data defined to say for which packet ID and fragment num, etc. we need to resend.

Personally, I believe this unnecessary complexity for the near term. He claims that some network infrastructure may consistently drop a particular packet if a stream is being sent. Do you have any experience with this? Any comments?

(VV) - I do not have any experience with this but am not sure that the ACK would solve the problem anyway. Is he saying that #3 would always get dropped if it is sent with the other fragments but would not get dropped if is sent alone. This does not sound right to me but I guess I do not know. It would be nice to not have to implement the notify. I will forward this to some people here to see if they have seen this type of a problem.

Thanks

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Thursday, January 03, 2002 7:40 AM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com
Subject: RE: 501 and frag comments

Hi Brian:

I read through the fragmentation proposal and overall it looks pretty good. The only thing I am not crazy about is the detection mechanism but the alternatives are worse. At least this way, only negotiations with packet loss are affected and not all negotiations. We could (and probably should) provide the user with a knob to set the IKE MTU size if the retries are taking too long or if the user knows ahead of time that there is a fragmentation issue with his NAT device.

Here are specific comments:

1. It looks like your MTU sizes are interface and not IKE based. IKE then needs to account for the MAC and IP headers accordingly so

that the IKE MTU values end up being around 1480 and 550.

2. In the FRAG1 and FRAG2 example, you have a total_frags value. total_frags is not in the payload definition, should it be? If it is part of the payload, do you still need the last fragment flag?

3. When you talk about allowing a single reassembly at a time, I am assuming you mean per peer? On the gateway side we have to allow reassembly for multiple peers simultaneously. This requirement should probably be relaxed.

4. It would be nice if the responder could learn the MTU size so that it would not have to go through the same autodetection procedure. This could be just a statement that says "If an ISA_FRAG payload is received, the receiver can assume that the IKE MTU can be no larger than the length of the received IKE packet. The receiver should update its MTU size if it is larger than the learned MTU."

5. For Continuous Channel Mode implementations, I think it would be OK to save the MTU information for rekeys. Again, this would save on the autodetection for every rekey.

I still need to go back and look at the 501 stuff.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Wednesday, January 02, 2002 5:54 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield
Subject: 501 and frag comments

Do you have any comments on this?

Still under NDA:

Apparently, we need to support 501. The current proposal, is to do to immediately float to 501 after detecting the NAT, and move the NAT-D payloads into the first payloads sent by the responder (SA payload). This is minimal effort.

The only issue I have with this approach is that it doesn't allow the peer outside the NAT to be stateless across MM rekeys. In a non-continuous channel MM approach, like we have, it is possible for all MMs to go away, and then for the external peer to need to rekey the MM. That peer will start on 500, which won't be allowed thru the NAT, and the connection will die unless the internal peer decides to rekey. This may be uncommon enough to not cause problems, but it might come back to bite us, too. That is my only concern.

The other option was:

Send an IP UDP ESP "ping" within IKE to detect the UDPESP unfriendly NATs: if it is detected, then either move to 501 to bypass the NAT, or allow the NAT to do the supposed ESP friendliness. This involves timeouts in the UDPESP unfriendly case.

bs

From: Brian Swander
Sent: Friday, January 04, 2002 10:02 AM
To: David Eitelbach; Rob Trace
Cc: Paul Mayfield
Subject: FW: 501 and frag comments

FYI: Have cisco agreement to the fragmentation proposal. Will now begin 501 discussions.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Friday, January 04, 2002 9:57 AM
To: Brian Swander; danyr@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com; Kartik Murthy
Subject: RE: 501 and frag comments

Yes, it looks like we all agree on our end.

Victor

> -----Original Message-----
> From: Brian Swander [mailto:briansw@windows.microsoft.com]
> Sent: Friday, January 04, 2002 12:54 PM
> To: Rochefort, Dany; Volpe, Victor
> Cc: William Dixon; Paul Mayfield; psd@cisco.com; Zubenko, Igor; Kartik Murthy
> Subject: RE: 501 and frag comments
>
> Are we agreed to this design? Any other outstanding issues?
>
> bs
>
> -----Original Message-----
> From: Dany Rochefort [mailto:danyr@cisco.com]
> Sent: Friday, January 04, 2002 9:22 AM
> To: Brian Swander; vvolpe@cisco.com
> Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com; Kartik Murthy
> Subject: RE: 501 and frag comments

Agreed. I was merely trying to protect the reassembly logic, but you bring up very good points that would minimize its usefulness.

-dany

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Friday, January 04, 2002 11:52 AM
To: Rochefort, Dany; Volpe, Victor
Cc: William Dixon; Paul Mayfield; psd@cisco.com; Zubenko, Igor; Kartik Murthy
Subject: RE: 501 and frag comments

This is extra complexity for little gain, IMHO. If we are fragmenting an encrypted payload (as is the common case for the ID payload), then all that can be modified is frag header fields. If they modify the encrypted data within the frag, that is identical to modifying an unfragmented, encrypted ID payload, which IKE is already robust against.

The worst this can do is cause the packet to fail reassembly and be lost, or scramble the packet during reassembly. Any attacker that can modify bits on the wire can already force IKE packets to be dropped anyway. Also, an attacker can similarily scramble the packet, encrypted or otherwise, and hashing/validation of the whole packet solves this (which we have once crypto is active).

Thus, I don't see how protecting the frag header is worth the effort. Also, then we'll have different semantics for before and after crypto keys are generated, and the problem gets much much tougher.

bs

-----Original Message-----
From: Dany Rochefort [mailto:danyr@cisco.com]
Sent: Friday, January 04, 2002 6:44 AM
To: Brian Swander; vvolpe@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com
Subject: RE: 501 and frag comments

Hi Brian,

Regarding the IKE_FRAG payload. In your proposal, you mention that

someone could modify some of the data since it's not encrypted. I was wondering if you had considered HASHING the IKE_FRAG itself to allow the peer to confirm that the IKE_FRAG is intact? I realize this does create some additional overhead.

-dany

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Thursday, January 03, 2002 6:12 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com; Zubenko, Igor
Subject: RE: 501 and frag comments

No, we don't need a containing payload, since we fragment the whole IKE packet, including the IKE hdr. So, the full blow up is:

Original non-fragmented packet: IKEHDR (Encrypt, nextp=ID) Data

becomes:

IKEHDR (Noencrypt, nextp=ISAFrag) IKEHDR(Encrypt,nextp=ID), beginning of data

IKEHDR (Noencrypt, nextp=ISAFrag) more data

Etc.

Thus, instead of defining the extra fields that we care about, (nextp and hdr flags at least) in the ISA_FRAG header, I thought it much simpler to just include the original hdr, too.

This solves your concern, doesn't it? Also, since the ISA_FRAG header isn't protected, you'd have to set these fields in every frag, validate they were the same each time, etc. This duplication just didn't seem worth it.

I don't see why you'd want to change to all FFs. I know all 0's in an invalid cookie in some RFC, and all FFs is probably valid, and I don't see what moving to all FFs would solve.

bs

-----Original Message-----

Exhibit B.txt

From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Thursday, January 03, 2002 1:59 PM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com; izubenko@cisco.com
Subject: RE: 501 and frag comments

Hi Brian:

One of our developers (Igor Zubenko on the CC list) has been implementing the fragmentation proposal and got it working but he ran into a few issues. One of the issues, the use of the total_frag value, we have already talked about. The other issue has to do with the ability to make the frag payload more generic. Since the contained payload value is overwritten in the IKE header, this will only work for MM pkts 5 and 6 (or for a single known first payload). Can you add a "contained_payload" field or something like that to the frag payload. This will remove the limitation and allow us to use it in different areas. For example, we are concerned that some transaction mode pkts are getting close to the MTU size. This will give us a way to get around that.

Also, Igor asked about the non-IKE marker in the port 500 version of NAT-T. His question is: why not use a cookie of all FFs. I know this was talked about early on but I do not remember the reasons why it was dismissed. Was it that all FFs is a valid cookie. If that was the only issue, NAT-T could specify that an all FF cookie has "special" meaning. Of course, there would still be the problem of IPsec aware NATs not changing the source port on IKE pkts.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Thursday, January 03, 2002 12:26 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com
Subject: RE: 501 and frag comments

Here's the info for the NAK:

Christian:

Don't bet on the 1/10 case. There is a quite common pattern in which fragment N is lost at every attempt. It happens for example if there is some kind of mistuned leaky bucket component if you are using IP over ATM. Please consider having at least a "NAK" of the form "please resend number N and following". Receiver should send the NAK if it receives an out of sequence packet.

bs

-----Original Message-----
From: Brian Swander
Sent: Wednesday, January 02, 2002 1:37 PM
To: Christian Huitema; Bernard Aboba; David Eitelbach; William Dixon; Ron Cully
Cc: Paul Mayfield
Subject: RE: Questions regarding the ipsec/NAT issue

Building in a NAK (or ACK) mechanism ala TCP doesn't seem essential for this release. Thus, I'd advocate not doing the partial ack for this release. Of course, IKE is extensible enough that this ACK scheme can be built in if necessary later. Indeed, if we write a draft on this, we can define an ACK scheme and make it optional to implement. If deployment experience shows it necessary, we can build it then.

Is this acceptable? I think it would be a very rare case where at lease one "fragment" is dropped on each of the retransmits. If we assume 576 MTU for IKE, a 5000 byte packet is approx 10 "fragments". We retransmit 3 times, and we'd have to have loss rates of 1/10 during each of these sends. Of course, not have an ACK scheme is bad for data transfers, but this should be ok for the IKE control traffic.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Thursday, January 03, 2002 9:15 AM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com
Subject: RE: 501 and frag comments

See below.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Thursday, January 03, 2002 11:59 AM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com
Subject: RE: 501 and frag comments

Answer to below:

1.Yes, I am speaking of the MTU for the interface. If you subtract the IP/UDP etc, then the MTUs for "IKE" will be smaller.

2.The example is wrong. Yes, we can do without the flags at the expense of the total_frags field in the hdr. However, that is less efficient to encode than a bit flag, and having flags around for extensibility can never hurt.

(VV) - I agree that the flags is a better way of handling this.

1.Each frag has a full IKE hdr. Thus, there will be a single reassembly per outstanding SA, which should be adequate even for a gateway, assuming that we are only fragmenting in MM, as is the case today. Of course, you can allow for multiple reassemblies per SA if you really want to. Remember, what you have is more an internal spec, not an internet draft. Thus, there are more internal implementation details than would commonly be in a draft.

(VV) - OK, that's what I thought but wanted to check.

4,5. Tradeoffs between complexity of internal state vs. messiness of the detection on the wire. We should make your proposed enhancements optional, since they won't effect interop.

(VV) - Sounds good.

Finally, (still under NDA): Christian is suggesting that we also build in a frag ack (NAK) to say: I didn't get frag #3, please resend it. This can simply be an unauthenticated notify with the data defined to say for which packet ID and fragment num, etc. we need to resend.

Personally, I believe this unnecessary complexity for the near term. He claims that some network infrastructure may consistently drop a particular packet if a stream is being sent. Do you have any experience with this? Any comments?

(VV) - I do not have any experience with this but am not sure that the ACK would solve the problem anyway. Is he saying that #3 would always get dropped if it is sent with the other fragments but would not get dropped if is sent alone. This does not sound right to me but I guess I do not know. It would be nice to not have to implement the notify. I will forward this to some people here to see if they have seen this type of a problem.

Thanks

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]

Sent: Thursday, January 03, 2002 7:40 AM
To: Brian Swander
Cc: William Dixon; Paul Mayfield;
danyr@cisco.com; psd@cisco.com
Subject: RE: 501 and frag comments

Hi Brian:

I read through the fragmentation proposal and overall it looks pretty good. The only thing I am not crazy about is the detection mechanism but the alternatives are worse. At least this way, only negotiations with packet loss are affected and not all negotiations. We could (and probably should) provide the user with a knob to set the IKE MTU size if the retries are taking too long or if the user knows ahead of time that there is a fragmentation issue with his NAT device.

Here are specific comments:

1. It looks like your MTU sizes are interface and not IKE based. IKE then needs to account for the MAC and IP headers accordingly so that the IKE MTU values end up being around 1480 and 550.

2. In the FRAG1 and FRAG2 example, you have a total_frags value. total_frags is not in the payload definition, should it be? If it is part of the payload, do you still need the last fragment flag?

3. When you talk about allowing a single reassembly at a time, I am assuming you mean per peer? On the gateway side we have to allow reassembly for multiple peers simultaneously. This requirement should probably be relaxed.

4. It would be nice if the responder could learn the MTU size so that it would not have to go through the same autodetection procedure. This could be just a statement that says "If an ISA_FRAG payload is received, the receiver can assume that the IKE MTU can be no larger than the length of the received IKE packet. The receiver should update its MTU size if it is larger than the learned MTU."

5. For Continuous Channel Mode implementations, I think it would be OK to save the MTU information for rekeys. Again, this would save on the autodetection for every rekey.

I still need to go back and look at the 501

stuff.

Victor

-----Original Message-----
From: Brian Swander
[mailto:briansw@windows.microsoft.com]
Sent: Wednesday, January 02, 2002 5:54 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield
Subject: 501 and frag comments

Do you have any comments on this?

Still under NDA:

Apparently, we need to support 501. The current proposal, is to do to immediately float to 501 after detecting the NAT, and move the NAT-D payloads into the first payloads sent by the responder (SA payload). This is minimal effort.

The only issue I have with this approach is that it doesn't allow the peer outside the NAT to be stateless across MM rekeys. In a non-continuous channel MM approach, like we have, it is possible for all MMs to go away, and then for the external peer to need to rekey the MM. That peer will start on 500, which won't be allowed thru the NAT, and the connection will die unless the internal peer decides to rekey. This may be uncommon enough to not cause problems, but it might come back to bite us, too. That is my only concern.

The other option was:

Send an IP UDP ESP "ping" within IKE to detect the UDPESP unfriendly NATs: if it is detected, then either move to 501 to bypass the NAT, or allow the NAT to do the supposed ESP friendliness. This involves timeouts in the UDPESP unfriendly case.

bs

From: Brian Swander
Sent: Friday, January 04, 2002 10:05 AM
To: 'Victor Volpe'; danyr@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com; Kartik Murthy
Subject: RE: 501 and frag comments

Excellent. One last time that I don't believe was in the spec you received. The actual vendorID payload to enable all this. I'm currently using a MD5 hash of FRAGMENTATION for the vendor id. Let me know if this is ok, or if some other string is more appealing.

It'd be nice to Interop this with you as soon as you get it running. Also, I don't

think we've had full Interop testing of the basic NAT traversal stuff, either. That's even more important asap.

Not to be pushy, but any chance to review the various 501 options? We are still assessing whether 501 is necessary or not, but current word is still that we need it.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Friday, January 04, 2002 9:57 AM
To: Brian Swander; danyr@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com; Kartik Murthy
Subject: RE: 501 and frag comments

Yes, it looks like we all agree on our end.

Victor

> -----Original Message-----
> From: Brian Swander [mailto:briansw@windows.microsoft.com]
> Sent: Friday, January 04, 2002 12:54 PM
> To: Rochefort, Dany; Volpe, Victor
> Cc: William Dixon; Paul Mayfield; psd@cisco.com; Zubenko, Igor; Kartik Murthy
> Subject: RE: 501 and frag comments
>
> Are we agreed to this design? Any other outstanding issues?
>
> bs
>
> -----Original Message-----
> From: Dany Rochefort [mailto:danyr@cisco.com]
> Sent: Friday, January 04, 2002 9:22 AM
> To: Brian Swander; vvolpe@cisco.com
> Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com; Kartik Murthy
> Subject: RE: 501 and frag comments
>
> Agreed. I was merely trying to protect the reassembly logic, but you bring up very good points that would minimize its usefulness.
>
> -dany

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Friday, January 04, 2002 11:52 AM
To: Rochefort, Dany; Volpe, Victor
Cc: William Dixon; Paul Mayfield; psd@cisco.com; Zubenko, Igor; Kartik Murthy
Subject: RE: 501 and frag comments

This is extra complexity for little gain, IMHO. If we are fragmenting an encrypted payload (as is the common case for the ID payload), then all that can be modified is frag header fields. If they modify the encrypted data within the frag, that is identical to modifying an unfragmented, encrypted ID payload, which IKE is already robust against.

The worst this can do is cause the packet to fail reassembly and be lost, or scramble the packet during reassembly. Any attacker that can modify bits on the wire can already force IKE packets to be dropped anyway. Also, an attacker can similarily scramble the packet, encrypted or otherwise, and hashing/validation of the whole packet solves this (which we have once crypto is active).

Thus, I don't see how protecting the frag header is worth the effort. Also, then we'll have different semantics for before and after crypto keys are generated, and the problem gets much much tougher.

bs

-----Original Message-----
From: Dany Rochefort [mailto:danyr@cisco.com]
Sent: Friday, January 04, 2002 6:44 AM
To: Brian Swander; vvolpe@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com
Subject: RE: 501 and frag comments

Hi Brian,

Regarding the IKE_FRAG payload. In your proposal, you mention that someone could modify some of the data since it's not encrypted. I was wondering if you had considered HASHING the IKE_FRAG itself to allow the peer to confirm that the IKE_FRAG is intact? I realize this does create some additional overhead.

-dany

-----Original Message-----

From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Thursday, January 03, 2002 6:12 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com; Zubenko, Igor
Subject: RE: 501 and frag comments

No, we don't need a containing payload, since we fragment the whole IKE packet, including the IKE hdr. So, the full blow up is:

Original non-fragmented packet: IKEHDR (Encrypt, nextp=ID) Data

becomes:

IKEHDR (Noencrypt, nextp=ISAFrag) IKEHDR(Encrypt,nextp=ID), beginning of data

IKEHDR (Noencrypt, nextp=ISAFrag) more data

Etc.

Thus, instead of defining the extra fields that we care about, (nextp and hdr flags at least) in the ISA_FRAG header, I thought it much simpler to just include the original hdr, too.

This solves your concern, doesn't it? Also, since the ISA_FRAG header isn't protected, you'd have to set these fields in every frag, validate they were the same each time, etc. This duplication just didn't seem worth it.

I don't see why you'd want to change to all FFs. I know all 0's in an invalid cookie in some RFC, and all FFs is probably valid, and I don't see what moving to all FFs would solve.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Thursday, January 03, 2002 1:59 PM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com; izubenko@cisco.com
Subject: RE: 501 and frag comments

Hi Brian:

One of our developers (Igor Zubenko on the CC list) has been implementing the fragmentation proposal and got it working but he ran into a few issues. One of the issues, the use of the total_frag value, we have already talked about. The other issue has to do with the ability to make the frag payload more generic. Since the contained payload value is overwritten in the IKE header, this will only work for MM pkts 5 and 6 (or for a single known first payload). Can you add a "contained_payload" field or something like that to the frag payload. This will remove the limitation and allow us to use it in different areas. For example, we are concerned that some transaction mode pkts are getting close to the MTU size. This will give us a way to get around that.

Also, Igor asked about the non-IKE marker in the port 500 version of NAT-T. His question is: why not use a cookie of all FFs. I know this was talked about early on but I do not remember the reasons why it was dismissed. Was it that all FFs is a valid cookie. If that was the only issue, NAT-T could specify that an all FF cookie has "special" meaning. Of course, there would still be the problem of IPsec aware NATs not changing the source port on IKE pkts.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Thursday, January 03, 2002 12:26 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com
Subject: RE: 501 and frag comments

Here's the info for the NAK:

Christian:

Don't bet on the 1/10 case. There is a quite common pattern in which fragment N is lost at every attempt. It happens for example if there is some kind of mistuned leaky bucket component if you are using IP over ATM. Please consider having at least a "NAK" of the form "please resend number N and following". Receiver should send the NAK if it receives an out of sequence packet.

bs

-----Original Message-----
From: Brian Swander
Sent: Wednesday, January 02, 2002 1:37 PM
To: Christian Huitema; Bernard Aboba; David Eitelbach; William Dixon; Ron Cully
Cc: Paul Mayfield
Subject: RE: Questions regarding the ipsec/NAT issue

Building in a NAK (or ACK) mechanism ala TCP doesn't seem essential for this release. Thus, I'd advocate not doing the partial ack for this release. Of course, IKE is extensible enough that this ACK scheme can be built in if necessary later. Indeed, if we write a draft on this, we can define an ACK scheme and make it optional to implement. If deployment experience shows it necessary, we can build it then.

Is this acceptable? I think it would be a very rare case where at lease one "fragment" is dropped on each of the retransmits. If we assume 576 MTU for IKE, a 5000 byte packet is approx 10 "fragments". We retransmit 3 times, and we'd have to have loss rates of 1/10 during each of these sends. Of course, not have an ACK scheme is bad for data transfers, but this should be ok for the IKE control traffic.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Thursday, January 03, 2002 9:15 AM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com
Subject: RE: 501 and frag comments

See below.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Thursday, January 03, 2002 11:59 AM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com
Subject: RE: 501 and frag comments

Answer to below:

1.Yes, I am speaking of the MTU for the interface. If you subtract the IP/UDP etc, then the MTUs for "IKE" will be smaller.

2.The example is wrong. Yes, we can do without the flags at the expense of the total_frags field in the hdr. However, that is less efficient to encode than a bit flag, and having flags around for extensibility can never hurt.

(VV) - I agree that the flags is a better way of handling this.

1.Each frag has a full IKE hdr. Thus, there will be a single reassembly per outstanding SA, which should be adequate even for a gateway, assuming that we are only fragmenting in MM, as is the case today. Of course, you can allow for multiple reassemblies per SA if you really want to. Remember, what you have is more an internal spec, not an internet draft. Thus, there are more internal implementation details than would commonly be in a draft.

(VV) - OK, that's what I thought but wanted to check.

4,5. Tradeoffs between complexity of internal state vs. messiness of the detection on the wire. We should make your proposed enhancements optional, since they won't effect interop.

(VV) - Sounds good.

Finally, (still under NDA): Christian is suggesting that we also build in a frag ack (NAK) to say: I didn't get frag #3, please resend it. This can simply be an unauthenticated notify with the data defined to say for which packet ID and fragment num, etc. we need to resend.

Personally, I believe this unnecessary complexity for the near term. He claims that some network infrastructure may consistently drop a particular packet if a stream is being sent. Do you have any experience with this? Any comments?

(VV) - I do not have any experience with this but am not sure that the ACK would solve the problem anyway. Is he saying that #3 would always get dropped if it is sent with the other fragments but would not get dropped if is sent alone. This does not sound right to me but I guess I do not know. It would be nice to not have to implement the notify. I will forward this to some people here to see if they have seen this type of a problem.

Thanks

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]

Sent: Thursday, January 03, 2002 7:40 AM
To: Brian Swander
Cc: William Dixon; Paul Mayfield;
danyr@cisco.com; psd@cisco.com
Subject: RE: 501 and frag comments

Hi Brian:

I read through the fragmentation proposal and overall it looks pretty good. The only thing I am not crazy about is the detection mechanism but the alternatives are worse. At least this way, only negotiations with packet loss are affected and not all negotiations. We could (and probably should) provide the user with a knob to set the IKE MTU size if the retries are taking too long or if the user knows ahead of time that there is a fragmentation issue with his NAT device.

Here are specific comments:

1. It looks like your MTU sizes are interface and not IKE based. IKE then needs to account for the MAC and IP headers accordingly so that the IKE MTU values end up being around 1480 and 550.

2. In the FRAG1 and FRAG2 example, you have a total_frags value. total_frags is not in the payload definition, should it be? If it is part of the payload, do you still need the last fragment flag?

3. When you talk about allowing a single reassembly at a time, I am assuming you mean per peer? On the gateway side we have to allow reassembly for multiple peers simultaneously. This requirement should probably be relaxed.

4. It would be nice if the responder could learn the MTU size so that it would not have to go through the same autodetection procedure. This could be just a statement that says "If an ISA_FRAG payload is received, the receiver can assume that the IKE MTU can be no larger than the length of the received IKE packet. The receiver should update its MTU size if it is larger than the learned MTU."

5. For Continuous Channel Mode implementations, I think it would be OK to save the MTU information for rekeys. Again, this would save on the autodetection for every rekey.

I still need to go back and look at the 501 stuff.

Victor

-----Original Message-----
From: Brian Swander
[mailto:briansw@windows.microsoft.com]

Exhibit B.txt
Sent: Wednesday, January 02, 2002 5:54 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield
Subject: 501 and frag comments

Do you have any comments on this?

Still under NDA:

Apparently, we need to support 501. The current proposal, is to do to immediately float to 501 after detecting the NAT, and move the NAT-D payloads into the first payloads sent by the responder (SA payload). This is minimal effort.

The only issue I have with this approach is that it doesn't allow the peer outside the NAT to be stateless across MM rekeys. In a non-continuous channel MM approach, like we have, it is possible for all MMs to go away, and then for the external peer to need to rekey the MM. That peer will start on 500, which won't be allowed thru the NAT, and the connection will die unless the internal peer decides to rekey. This may be uncommon enough to not cause problems, but it might come back to bite us, too. That is my only concern.

The other option was:

Send an IP UDP ESP "ping" within IKE to detect the UDPESP unfriendly NATs: if it is detected, then either move to 501 to bypass the NAT, or allow the NAT to do the supposed ESP friendliness. This involves timeouts in the UDPESP unfriendly case.

bs

From: Brian Swander
Sent: Friday, January 04, 2002 12:28 PM
To: 'Igor Zubenko'
Subject: RE: 501 and frag comments

ISA_FRAG = 132

Currently, we'll only send or receive frags for the ID payload, since that is the only known problem thus far. The only message I've seen getting close to 576 is the MM SA payload because of the vendor ID blowup, but that is still below 576 in all normal scenarios.

I wanted to keep the code change and possible risk of attack absolutely minimal instead of going for the larger, general solution.

bs

-----Original Message-----
From: Igor Zubenko [mailto:izubenko@cisco.com]
Sent: Friday, January 04, 2002 11:49 AM
To: Brian Swander
Subject: RE: 501 and frag comments

Hi Brian,

There are a few more questions about IKE fragmentation:

1) What is the value of ISA_FRAG?

2) If we are going to support the MTU size of 576 for Phase I messages 5 and 6, does it mean that we also have to fragment other messages within the negotiation that are above 576 bytes? Currently I've seen such messages within Phases I and II.

Thank you,

Igor Zubenko

> -----Original Message-----
> From: Brian Swander [mailto:briansw@windows.microsoft.com]
> Sent: Friday, January 04, 2002 1:05 PM
> To: Volpe, Victor; Rochefort, Dany
> Cc: William Dixon; Paul Mayfield; psd@cisco.com; Zubenko, Igor; Kartik Murthy
> Subject: RE: 501 and frag comments
>
> Excellent. One last time that I don't believe was in the spec you received. The actual vendorID payload to enable all this. I'm currently using a MD5 hash of FRAGMENTATION for the vendor id. Let me know if this is ok, or if some other string is more appealing.
>
> It'd be nice to Interop this with you as soon as you get it running. Also, I don't think we've had full Interop testing of the basic NAT traversal stuff, either. That's even more important asap.
>
> Not to be pushy, but any chance to review the various 501 options? We are still assessing whether 501 is necessary or not, but current word is still that we need it.
>
> bs
>
> -----Original Message-----

From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Friday, January 04, 2002 9:57 AM
To: Brian Swander; danyr@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com; Kartik Murthy
Subject: RE: 501 and frag comments

Yes, it looks like we all agree on our end.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Friday, January 04, 2002 12:54 PM
To: Rochefort, Dany; Volpe, Victor
Cc: William Dixon; Paul Mayfield; psd@cisco.com; Zubenko, Igor; Kartik Murthy
Subject: RE: 501 and frag comments

Are we agreed to this design? Any other outstanding issues?

bs

-----Original Message-----
From: Dany Rochefort [mailto:danyr@cisco.com]
Sent: Friday, January 04, 2002 9:22 AM
To: Brian Swander; vvolpe@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com; Kartik Murthy
Subject: RE: 501 and frag comments

Agreed. I was merely trying to protect the reassembly logic, but you bring up very good points that would minimize its usefulness.

-dany

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Friday, January 04, 2002 11:52 AM
To: Rochefort, Dany; Volpe, Victor
Cc: William Dixon; Paul Mayfield; psd@cisco.com; Zubenko, Igor; Kartik Murthy
Subject: RE: 501 and frag comments

This is extra complexity for little gain, IMHO. If we are fragmenting an encrypted payload (as is the common case for the ID payload), then all that can be modified is frag header fields. If they modify the encrypted data within the frag, that is identical to modifying an unfragmented, encrypted ID payload, which IKE is already robust against.

The worst this can do is cause the packet to fail reassembly and be lost, or scramble the packet during reassembly. Any attacker that can modify bits on the wire can already force IKE packets to be dropped anyway. Also, an attacker can similarily scramble the packet, encrypted or otherwise, and hashing/validation of the whole packet solves this (which we have once crypto is active).

Thus, I don't see how protecting the frag header is worth the effort. Also, then we'll have different semantics for before and after crypto keys are generated, and the problem gets much much tougher.

bs

-----Original Message-----
From: Dany Rochefort [mailto:danyr@cisco.com]
Sent: Friday, January 04, 2002 6:44 AM
To: Brian Swander; vvolpe@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com
Subject: RE: 501 and frag comments

Hi Brian,

Regarding the IKE_FRAG payload. In your proposal, you mention that someone could modify some of the data since it's not encrypted. I was wondering if you had considered HASHING the IKE_FRAG itself to allow the peer to confirm that the IKE_FRAG is intact? I realize this does create some additional overhead.

-dany

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Thursday, January 03, 2002 6:12 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com; Zubenko, Igor
Subject: RE: 501 and frag comments

No, we don't need a containing payload, since we fragment the whole IKE packet, including the IKE hdr. So, the full blow up is:

Original non-fragmented packet: IKEHDR (Encrypt, nextp=ID) Data

becomes:

IKEHDR (Noencrypt, nextp=ISAFrag)
IKEHDR(Encrypt,nextp=ID), beginning of data

IKEHDR (Noencrypt, nextp=ISAFrag) more data

Etc.

Thus, instead of defining the extra fields that we care about, (nextp and hdr flags at least) in the ISA_FRAG header, I thought it much simpler to just include the original hdr, too.

This solves your concern, doesn't it? Also, since the ISA_FRAG header isn't protected, you'd have to set these fields in every frag, validate they were the same each time, etc. This duplication just didn't seem worth it.

I don't see why you'd want to change to all FFs. I know all 0's in an invalid cookie in some RFC, and all FFs is probably valid, and I don't see what moving to all FFs would solve.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Thursday, January 03, 2002 1:59 PM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com; izubenko@cisco.com
Subject: RE: 501 and frag comments

Hi Brian:

One of our developers (Igor Zubenko on the CC list) has been implementing the fragmentation proposal and got it working but he ran into a few issues. One of the issues, the use of the total_frag value, we have already talked about. The other issue has to do with the ability to make the frag payload more generic. Since the contained payload value is overwritten in the IKE header, this will only work for MM pkts 5 and 6 (or for a single known first payload). Can you add a "contained_payload" field or something like that to the frag payload.

This will remove the limitation and allow us to use it in different areas. For example, we are concerned that some transaction mode pkts are getting close to the MTU size. This will give us a way to get around that.

Also, Igor asked about the non-IKE marker in the port 500 version of NAT-T. His question is: why not use a cookie of all FFs. I know this was talked about early on but I do not remember the reasons why it was dismissed. Was it that all FFs is a valid cookie. If that was the only issue, NAT-T could specify that an all FF cookie has "special" meaning. Of course, there would still be the problem of IPsec aware NATs not changing the source port on IKE pkts.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Thursday, January 03, 2002 12:26 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com
Subject: RE: 501 and frag comments

Here's the info for the NAK:

Christian:

Don't bet on the 1/10 case. There is a quite common pattern in which fragment N is lost at every attempt. It happens for example if there is some kind of mistuned leaky bucket component if you are using IP over ATM. Please consider having at least a "NAK" of the form "please resend number N and following". Receiver should send the NAK if it receives an out of sequence packet.

bs

-----Original Message-----
From: Brian Swander
Sent: Wednesday, January 02, 2002 1:37 PM
To: Christian Huitema; Bernard Aboba; David Eitelbach; William Dixon; Ron Cully
Cc: Paul Mayfield
Subject: RE: Questions regarding the ipsec/NAT issue

Building in a NAK (or ACK) mechanism ala TCP doesn't seem essential for this release. Thus, I'd advocate not doing the partial ack for this release. Of course, IKE is extensible enough that this ACK scheme can be built in if necessary later. Indeed, if we write a draft on this, we can define an ACK scheme and make it optional to implement. If deployment experience shows it necessary, we can build it then.

Is this acceptable? I think it would be a very rare case where at lease one “fragment” is dropped on each of the retransmits. If we assume 576 MTU for IKE, a 5000 byte packet is approx 10 “fragments”. We retransmit 3 times, and we’d have to have loss rates of 1/10 during each of these sends. Of course, not have an ACK scheme is bad for data transfers, but this should be ok for the IKE control traffic.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]

Sent: Thursday, January 03, 2002 9:15 AM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com
Subject: RE: 501 and frag comments

See below.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Thursday, January 03, 2002 11:59 AM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com
Subject: RE: 501 and frag comments

Answer to below:

1.Yes, I am speaking of the MTU for the interface. If you subtract the IP/UDP etc, then the MTUs for “IKE” will be smaller.

2.The example is wrong. Yes, we can do without the flags at the expense of the total_frags field in the hdr. However, that is less efficient to encode than a bit flag, and having flags around for extensibility can never hurt.

(VV) - I agree that the flags is a better way of handling this.

1.Each frag has a full IKE hdr. Thus, there will be a single reassembly per outstanding SA, which should be adequate even for a

gateway, assuming that we are only fragmenting in MM, as is the case today. Of course, you can allow for multiple reassemblies per SA if you really want to. Remember, what you have is more an internal spec, not an internet draft. Thus, there are more internal implementation details than would commonly be in a draft.

(VV) - OK, that's what I thought but wanted to check.

4,5. Tradeoffs between complexity of internal state vs. messiness of the detection on the wire. We should make your proposed enhancements optional, since they won't effect interop.

(VV) - Sounds good.

Finally, (still under NDA): Christian is suggesting that we also build in a frag ack (NAK) to say: I didn't get frag #3, please resend it. This can simply be an unauthenticated notify with the data defined to say for which packet ID and fragment num, etc. we need to resend.

Personally, I believe this unnecessary complexity for the near term. He claims that some network infrastructure may consistently drop a particular packet if a stream is being sent. Do you have any experience with this? Any comments?

(VV) - I do not have any experience with this but am not sure that the ACK would solve the problem anyway. Is he saying that #3 would always get dropped if it is sent with the other fragments but would not get dropped if is sent alone. This does not sound right to me but I guess I do not know. It would be nice to not have to implement the notify. I will forward this to some people here to see if they have seen this type of a problem.

Thanks

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]

Sent: Thursday, January 03, 2002 7:40 AM
To: Brian Swander
Cc: William Dixon; Paul Mayfield;
danyr@cisco.com; psd@cisco.com

Subject: RE: 501 and frag comments

Hi Brian:

I read through the fragmentation proposal and overall it looks pretty good. The only thing I am not crazy about is the detection mechanism but the alternatives are worse. At least this way, only negotiations with packet loss are affected and not all negotiations. We could (and probably should) provide the user with a knob to set the IKE MTU size if the retries are taking too long or if the user knows ahead of time that there is a fragmentation issue with his NAT device.

Here are specific comments:

1. It looks like your MTU sizes are interface and not IKE based. IKE then needs to account for the MAC and IP headers accordingly so that the IKE MTU values end up being around 1480 and 550.

2. In the FRAG1 and FRAG2 example, you have a total_frags value. total_frags is not in the payload definition, should it be? If it is part of the payload, do you still need the last fragment flag?

3. When you talk about allowing a single reassembly at a time, I am assuming you mean per peer? On the gateway side we have to allow reassembly for multiple peers simultaneously. This requirement should probably be relaxed.

4. It would be nice if the responder could learn the MTU size so that it would not have to go through the same autodetection procedure. This could be just a statement that says "If an ISA_FRAG payload is received, the receiver can assume that the IKE MTU can be no larger than the length of the received IKE packet. The receiver should update its MTU size if it is larger than the learned MTU."

5. For Continuous Channel Mode implementations, I think it would be OK to save the MTU information for rekeys. Again, this would save on the autodetection for every rekey.

I still need to go back and look at the 501 stuff.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Wednesday, January 02, 2002 5:54 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield
Subject: 501 and frag comments

Do you have any comments on this?

Still under NDA:

Apparently, we need to support 501. The current proposal, is to do to immediately float to 501 after detecting the NAT, and move the NAT-D payloads into the first payloads sent by the responder (SA payload). This is minimal effort.

The only issue I have with this approach is that it doesn't allow the peer outside the NAT to be stateless across MM rekeys. In a non-continuous channel MM approach, like we have, it is possible for all MMs to go away, and then for the external peer to need to rekey the MM. That peer will start on 500, which won't be allowed thru the NAT, and the connection will die unless the internal peer decides to rekey. This may be uncommon enough to not cause problems, but it might come back to bite us, too. That is my only concern.

The other option was:

Send an IP UDP ESP "ping" within IKE to detect the UDPESP unfriendly NATs: if it is detected, then either move to 501 to bypass the NAT, or allow the NAT to do the supposed ESP friendliness. This involves timeouts in the UDPESP unfriendly case.

bs

From: Brian Swander
Sent: Friday, January 04, 2002 1:54 PM
To: Kartik Murthy
Subject: ikefrag

bs

From: Brian Swander
Sent: Friday, January 04, 2002 1:57 PM
To: 'Victor Volpe'; danyr@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com; Kartik Murthy
Subject: RE: 501 and frag comments

The best I've got is this from William earlier:

In looking at these again, and after testing NATs, both 2 and 3 assume that IPSec passthru NATs are implemented well, which is a bad assumption. We have seen that many only allow 1 connection thru in passthru mode. Thus, only #1 allows multiple connections thru any NAT.

1. Start on 500, discover the NAT, and switch to a new, non-500 port. After much hair pulling, Brian and I propose the following:

```
UDP src 500, dst 500  [HDR][SA][NAT-T VENDOR ID] ---->

                      <--- [HDR][SA][NAT-T VENDOR ID][NAT-D]

UDP src 500, dst 501  [HDR][SA][NAT-T VENDOR ID][NAT-D] --->

                      <--- [HDR][SA][NAT-T VENDOR ID][NAT-D]
```

This approach moves NAT-D to the responder in the case where NAT-T capability is detected. This avoids backward compat problems with normal IKE initiators. It adds the expense of including a NAT-D payload in every response to a NAT-T capable initiator. The new NAT-T initiator then re-initiates to the 501 port, and also includes NAT-D in the first exchange. We did this to avoid NAT-D during the KE exchange to avoid generating a DH and then discovering we have to re-initiate to move the port. This approach continues to use UDP-ESP, though with the more efficient UDP-ESP encapsulation, which will make many people happy. It doesn't increase RTs. Downside, is that it is a NAT-T draft change, and change to how anyone building ESP-UDP probably implemented already (with 0x00). Keep-alive is done for 501 only in this case to the dest IP. Firewall admins have to open 501 in addition to 501.

2. Add in MM a ping to detect the "IPSec passthrough" NAT mode.

```
UDP src 500, dst 500  [HDR:I-COOKIE=0x00:R-COOKIE=0x00][VENDORID NAT-T] ---->

                      <--- [HDR:I-COOKIE=0x00:R-COOKIE=0x02][VENDORID NAT-T][NAT-D]
```

This stateless ping could happen as the very first packet, but then you'd always ping even before you knew there was a NAT. This might be helpful anyway if merged with the IPSec SA keep-alive somehow. And with NAT-D in the response, you can tell whether there is a NAT. If you added this ping after MM completed, then you'd know there was a NAT and you're just testing to see if UDP-ESP with 0x00 will get through. Note that the R-COOKIE must not be the same as the R-COOKIE of the outbound packet to make this test valid.

3. IKE requests a new QM IPSec SA with a normal [IP][ESP] encapsulation, not UDP-ESP 0x00.

IKE completes MM, detects NAT, and then decides a default (or configured) way to establish the IPSec SA in QM - picking either [IP][ESP] or [IP][UDP][0x00][ESP]. If it's default, you guess which method is best for your deployment. If it's configured, it's because you know what kind of NAT you need to go through. It's possible that if 0bytes are received on the inbound SA, then you could automatically redo a QM to propose the other method. IKE implementations would have to be able to rekey QM with either proposal set after NAT is detected. Upper layer protocols that were IPsec aware, could also be aware that their initial connect was not getting through, and to request the alternate IPSec SA type.

-----Original Message-----

From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Friday, January 04, 2002 12:36 PM
To: Brian Swander; danyr@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com; Kartik Murthy
Subject: RE: 501 and frag comments

Brian:

I have been looking through the threads on port 501 but they are all pretty schetchy. Do you have anything that spells everything out in one doc?

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Friday, January 04, 2002 1:05 PM
To: Volpe, Victor; Rochefort, Dany
Cc: William Dixon; Paul Mayfield; psd@cisco.com; Zubenko, Igor; Kartik Murthy
Subject: RE: 501 and frag comments

Excellent. One last time that I don't believe was in the spec you received. The actual vendorID payload to enable all this. I'm currently using a MD5 hash of FRAGMENTATION for the vendor id. Let me know if this is ok, or if some other string is more appealing.

It'd be nice to Interop this with you as soon as you get it running. Also, I don't think we've had full Interop testing of the basic NAT traversal stuff, either. That's even more important asap.

Not to be pushy, but any chance to review the various 501 options? We are still assessing whether 501 is necessary or not, but current word is still that we need it.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Friday, January 04, 2002 9:57 AM
To: Brian Swander; danyr@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com; Kartik Murthy
Subject: RE: 501 and frag comments

Yes, it looks like we all agree on our end.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Friday, January 04, 2002 12:54 PM
To: Rochefort, Dany; Volpe, Victor
Cc: William Dixon; Paul Mayfield; psd@cisco.com; Zubenko, Igor; Kartik Murthy
Subject: RE: 501 and frag comments

Are we agreed to this design? Any other outstanding issues?

bs

-----Original Message-----
From: Dany Rochefort [mailto:danyr@cisco.com]
Sent: Friday, January 04, 2002 9:22 AM
To: Brian Swander; vvolpe@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com; Kartik Murthy
Subject: RE: 501 and frag comments

Agreed. I was merely trying to protect the reassembly logic, but you bring up very good points that would minimize its usefulness.

-dany

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Friday, January 04, 2002 11:52 AM
To: Rochefort, Dany; Volpe, Victor
Cc: William Dixon; Paul Mayfield; psd@cisco.com; Zubenko, Igor; Kartik Murthy
Subject: RE: 501 and frag comments

This is extra complexity for little gain, IMHO. If we are fragmenting an encrypted payload (as is the common case for the ID payload), then all that can be modified is frag header fields. If they modify the encrypted data within the frag, that is identical to modifying an unfragmented, encrypted ID payload, which IKE is already robust against.

The worst this can do is cause the packet to fail reassembly and be lost, or scramble the packet during reassembly. Any attacker that can modify bits on the wire can already force IKE packets to be dropped anyway. Also, an attacker can similarily scramble the packet, encrypted or otherwise, and hashing/validation of the whole packet solves this (which we have once crypto is active).

Thus, I don't see how protecting the frag header is worth the effort. Also, then we'll have different semantics for before and after crypto

keys are generated, and the problem gets much much tougher.

bs

-----Original Message-----
From: Dany Rochefort [mailto:danyr@cisco.com]
Sent: Friday, January 04, 2002 6:44 AM
To: Brian Swander; vvolpe@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com
Subject: RE: 501 and frag comments

Hi Brian,

Regarding the IKE_FRAG payload. In your proposal, you mention that someone could modify some of the data since it's not encrypted. I was wondering if you had considered HASHING the IKE_FRAG itself to allow the peer to confirm that the IKE_FRAG is intact? I realize this does create some additional overhead.

-dany

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Thursday, January 03, 2002 6:12 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com; Zubenko, Igor
Subject: RE: 501 and frag comments

No, we don't need a containing payload, since we fragment the whole IKE packet, including the IKE hdr. So, the full blow up is:

Original non-fragmented packet: IKEHDR (Encrypt, nextp=ID) Data

becomes:

IKEHDR (Noencrypt, nextp=ISAFrag) IKEHDR(Encrypt,nextp=ID), beginning of data

IKEHDR (Noencrypt, nextp=ISAFrag) more data

Etc.

Thus, instead of defining the extra fields that we care about, (nextp and hdr flags at least) in the ISA_FRAG header, I thought it much simpler to just include the original hdr, too.

This solves your concern, doesn't it? Also, since the ISA_FRAG header isn't protected, you'd have to set these fields in every frag, validate they were the same each time, etc. This duplication just didn't seem worth it.

I don't see why you'd want to change to all FFs. I know all 0's in an invalid cookie in some RFC, and all FFs is probably valid, and I don't see what moving to all FFs would solve.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Thursday, January 03, 2002 1:59 PM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com; izubenko@cisco.com
Subject: RE: 501 and frag comments

Hi Brian:

One of our developers (Igor Zubenko on the CC list) has been implementing the fragmentation proposal and got it working but he ran into a few issues. One of the issues, the use of the total_frag value, we have already talked about. The other issue has to do with the ability to make the frag payload more generic. Since the contained payload value is overwritten in the IKE header, this will only work for MM pkts 5 and 6 (or for a single known first payload). Can you add a "contained_payload" field or something like that to the frag payload. This will remove the limitation and allow us to use it in different areas. For example, we are concerned that some transaction mode pkts are getting close to the MTU size. This will give us a way to get around that.

Also, Igor asked about the non-IKE marker in the port 500 version of NAT-T. His question is: why not use a cookie of all FFs. I know this was talked about early on but I do not remember the reasons why it was dismissed. Was it that all FFs is a valid cookie. If that was the only issue, NAT-T could specify that an all FF cookie has "special" meaning. Of course, there would still be the problem of IPsec aware NATs not changing the source port on IKE

pkts.

Victor

-----Original Message-----
From: Brian Swander
[mailto:briansw@windows.microsoft.com]
Sent: Thursday, January 03, 2002 12:26 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com
Subject: RE: 501 and frag comments

Here's the info for the NAK:

Christian:

Don't bet on the 1/10 case. There is a quite common pattern in which fragment N is lost at every attempt. It happens for example if there is some kind of mistuned leaky bucket component if you are using IP over ATM. Please consider having at least a "NAK" of the form "please resend number N and following". Receiver should send the NAK if it receives an out of sequence packet.

bs

-----Original Message-----
From: Brian Swander
Sent: Wednesday, January 02, 2002 1:37 PM
To: Christian Huitema; Bernard Aboba; David Eitelbach; William Dixon; Ron Cully
Cc: Paul Mayfield
Subject: RE: Questions regarding the ipsec/NAT issue

Building in a NAK (or ACK) mechanism ala TCP doesn't seem essential for this release. Thus, I'd advocate not doing the partial ack for this release. Of course, IKE is extensible enough that this ACK scheme can be built in if necessary later. Indeed, if we write a draft on this, we can define an ACK scheme and make it optional to implement. If deployment experience shows it necessary, we can build it then.

Is this acceptable? I think it would be a very rare case where at lease one "fragment" is dropped on each of the retransmits. If we assume 576 MTU for IKE, a 5000 byte packet is approx 10 "fragments". We retransmit 3 times, and we'd have to have loss rates of 1/10 during each of these sends. Of course, not have an ACK scheme is bad for data transfers, but this should be ok for the IKE control traffic.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]

Sent: Thursday, January 03, 2002 9:15 AM
To: Brian Swander
Cc: William Dixon; Paul Mayfield;
danyr@cisco.com; psd@cisco.com
Subject: RE: 501 and frag comments

See below.

Victor

-----Original Message-----
From: Brian Swander
[mailto:briansw@windows.microsoft.com]

Sent: Thursday, January 03, 2002 11:59 AM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort,
Dany; psd@cisco.com
Subject: RE: 501 and frag comments

Answer to below:

1.Yes, I am speaking of the MTU for the interface. If you subtract the IP/UDP etc, then the MTUs for "IKE" will be smaller.

2.The example is wrong. Yes, we can do without the flags at the expense of the total_frags field in the hdr. However, that is less efficient to encode than a bit flag, and having flags around for extensibility can never hurt.

(VV) - I agree that the flags is a better way of handling this.

1.Each frag has a full IKE hdr. Thus, there will be a single reassembly per outstanding SA, which should be adequate even for a gateway, assuming that we are only fragmenting in MM, as is the case today. Of course, you can allow for multiple reassemblies per SA if you really want to. Remember, what you have is more an internal spec, not an internet draft. Thus, there are more internal implementation details than would commonly be in a draft.

(VV) - OK, that's what I thought but wanted to check.

4,5. Tradeoffs between complexity of internal state vs. messiness of the detection on the wire. We should make your proposed enhancements optional, since they won't effect interop.

(VV) - Sounds good.

Finally, (still under NDA): Christian is suggesting that we also build in a frag ack (NAK) to say: I didn't get frag #3, please resend it. This can simply be an unauthenticated notify with the data defined to say for which packet ID and fragment num, etc. we need to resend.

Personally, I believe this unnecessary complexity for the near term. He claims that some network infrastructure may consistently drop a particular packet if a stream is being sent. Do you have any experience with this? Any comments?

(VV) - I do not have any experience with this but am not sure that the ACK would solve the problem anyway. Is he saying that #3 would always get dropped if it is sent with the other fragments but would not get dropped if is sent alone. This does not sound right to me but I guess I do not know. It would be nice to not have to implement the notify. I will forward this to some people here to see if they have seen this type of a problem.

Thanks

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]

Sent: Thursday, January 03, 2002 7:40 AM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com
Subject: RE: 501 and frag comments

Hi Brian:

I read through the fragmentation proposal and overall it looks pretty good. The only thing I am not crazy about is the detection mechanism but the alternatives are worse. At least this way, only negotiations with packet loss are affected and not all negotiations. We could (and probably should) provide the user with a knob to set the IKE MTU size if the retries are taking too long or if the user knows ahead of time that there is a fragmentation issue with his NAT device.

Here are specific comments:

1. It looks like your MTU sizes are interface and not IKE based. IKE then needs to account for the MAC and IP headers accordingly so that the IKE MTU values end up being around 1480 and 550.

2. In the FRAG1 and FRAG2 example, you have a total_frags value. total_frags is not in the payload definition, should it be? If it is part of the payload, do you still need the last fragment flag?

3. When you talk about allowing a single reassembly at a time, I am assuming you mean per peer? On the gateway side we have to allow reassembly for multiple peers simultaneously. This requirement should probably be relaxed.

4. It would be nice if the responder could learn the MTU size so that it would not have to go through the same autodetection procedure. This could be just a statement that says "If an ISA_FRAG payload is received, the receiver can assume that the IKE MTU can be no larger than the length of the received IKE packet. The receiver should update its MTU size if it is larger than the learned MTU."

5. For Continuous Channel Mode implementations, I think it would be OK to save the MTU information for rekeys. Again, this would save on the autodetection for every rekey.

I still need to go back and look at the 501 stuff.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Wednesday, January 02, 2002 5:54 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield
Subject: 501 and frag comments

Do you have any comments on this?

Still under NDA:

Apparently, we need to support 501. The current proposal, is to do to immediately float to 501 after detecting the NAT, and move the NAT-D payloads into the first payloads sent by the responder (SA payload). This is minimal effort.

The only issue I have with this approach is that it doesn't allow the peer outside the NAT to be stateless across MM rekeys.

In a non-continuous channel MM approach, like we have, it is possible for all MMs to go away, and then for the external peer to need to rekey the MM. That peer will start on 500, which won't be allowed thru the NAT, and the connection will die unless the internal peer decides to rekey. This may be uncommon enough to not cause problems, but it might come back to bite us, too. That is my only concern.

The other option was:

Send an IP UDP ESP "ping" within IKE to detect the UDPESP unfriendly NATs: if it is detected, then either move to 501 to bypass the NAT, or allow the NAT to do the supposed ESP friendliness. This involves timeouts in the UDPESP unfriendly case.

bs

From: Brian Swander
Sent: Friday, January 04, 2002 4:10 PM
To: 'Victor Volpe'; danyr@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com; Kartik Murthy
Subject: RE: 501 and frag comments

The proposal below is inaccurate and a little old. The plan is really to do the deterministic switch to 501 after the first 2 pkts.

```
UDP src 500, dst 500  [HDR][SA][NAT-T VENDOR ID] ---->

                      <--- [HDR][SA][NAT-T VENDOR ID][NAT-D]

UDP src 501, dst 501  [HDR][KE][NAT-D] --->

                      <--- [HDR][KE]
```

Src port of 500 I believe is an oversight in William's notes. Once you float to 501, to should float both src and dst.

Answers to below:

1. There is no delete needed since there is only 1 SA brought up.

2. Pkt loss just handled via standard retransmission scheme. Initiator will just keep sending on 501, and if no response, the connection dies

3. Yes, you should support this if you have state and know that the peer at that addr supports 501.

4. Yes, there are cornercase rekey issues. We have chosen to ignore them as likely minimal, but we can spec solutions. The scenario is:

C NAT X(external)

Now, we get an SA between C and X. Now, if C rekeys the MM, it is ok, since the NAT will either reuse the ports or allocate a new one. If X rekeys the MM and has state to remember to use 501, it'll be ok, since the NAT will still have a 501 mapping to allow the new request in. However, if the rare cases where X deletes all state, and then needs to rekey the MM, X would initiate on 500, and the NAT will not allow it in. Of course, the fix is to make sure that X remembers to use 501, but this is tougher to implement.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Friday, January 04, 2002 2:14 PM
To: Brian Swander; danyr@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com; Kartik Murthy
Subject: RE: 501 and frag comments

I did re-look at #1 earlier but it is really not enough to comment on. There are some things I like about it:

1. The deterministic switch to port 501 after the first 2 pkts

2. The fact that the Initiator is responsible for reinitiating from the new port

There are some things that I do not fully understand. The main issue is why a source port of 500 is still used once the switch to 501 is made?

We need to answer these questions as well:

1. Is there a delete message sent when the switch takes place or is the delete implicit?

2. What happens when there is pkt loss during the switch?

3. Can implementations accept IKE on port 501 without it starting on 500?

4. Rekey issues?

Sorry for asking for this but can you put together a more detailed description of how this is all going to work. I think a lot of this was written down long ago when you guys had your initial port 501 proposal. Do you still have the notes from that initial meeting?

I am leaving in about 10 minutes so if you do not hear back from me that is why.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Friday, January 04, 2002 4:57 PM
To: Volpe, Victor; Rochefort, Dany
Cc: William Dixon; Paul Mayfield; psd@cisco.com; Zubenko, Igor; Kartik Murthy
Subject: RE: 501 and frag comments

The best I've got is this from William earlier:

In looking at these again, and after testing NATs, both 2 and 3 assume that IPSec passthru NATs are implemented well, which is a bad assumption. We have seen that many only allow 1 connection thru in passthru mode. Thus, only #1 allows multiple connections thru any NAT.

1. Start on 500, discover the NAT, and switch to a new, non-500 port. After much hair pulling, Brian and I propose the following:

```
UDP src 500, dst 500  [HDR][SA][NAT-T VENDOR ID] ---->
                      <--- [HDR][SA][NAT-T VENDOR ID][NAT-D]
UDP src 500, dst 501  [HDR][SA][NAT-T VENDOR ID][NAT-D] --->
                      <--- [HDR][SA][NAT-T VENDOR ID][NAT-D]
```

This approach moves NAT-D to the responder in the case where NAT-T capability is detected. This avoids backward compat problems with normal IKE initiators. It adds the expense of including a NAT-D payload in every response to a NAT-T capable initiator. The new NAT-T initiator then re-initiates to the 501 port, and also includes NAT-D in the first exchange. We did this to avoid NAT-D during the KE exchange to avoid generating a DH and then discovering we have to re-initiate to move the port. This approach continues to use UDP-ESP, though with the more efficient UDP-ESP encapsulation, which will make many people happy. It doesn't increase RTs. Downside, is that it is a NAT-T draft change, and change to how anyone building ESP-UDP probably implemented already (with 0x00). Keep-alive is done for 501 only in this case to the dest IP. Firewall admins have to open 501 in addition to 501.

2. Add in MM a ping to detect the "IPSec passthrough" NAT mode.

UDP src 500, dst 500 [HDR:I-COOKIE=0x00:R-COOKIE=0x00][VENDORID NAT-T] ---->

<--- [HDR:I-COOKIE=0x00:R-COOKIE=0x02][VENDORID NAT-T][NAT-D]

This stateless ping could happen as the very first packet, but then you'd always ping even before you knew there was a NAT. This might be helpful anyway if merged with the IPSec SA keep-alive somehow. And with NAT-D in the response, you can tell whether there is a NAT. If you added this ping after MM completed, then you'd know there was a NAT and you're just testing to see if UDP-ESP with 0x00 will get through. Note that the R-COOKIE must not be the same as the R-COOKIE of the outbound packet to make this test valid.

3. IKE requests a new QM IPSec SA with a normal [IP][ESP] encapsulation, not UDP-ESP 0x00.

IKE completes MM, detects NAT, and then decides a default (or configured) way to establish the IPSec SA in QM - picking either [IP][ESP] or [IP][UDP][0x00][ESP]. If it's default, you guess which method is best for your deployment. If it's configured, it's because you know what kind of NAT you need to go through. It's possible that if 0bytes are received on the inbound SA, then you could automatically redo a QM to propose the other method. IKE implementations would have to be able to rekey QM with either proposal set after NAT is detected. Upper layer protocols that were IPsec aware, could also be aware that their initial connect was not getting through, and to request the alternate IPSec SA type.

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Friday, January 04, 2002 12:36 PM
To: Brian Swander; danyr@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com; Kartik Murthy
Subject: RE: 501 and frag comments

Brian:

I have been looking through the threads on port 501 but they are all pretty schetchy. Do you have anything that spells everything out in one doc?

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Friday, January 04, 2002 1:05 PM
To: Volpe, Victor; Rochefort, Dany
Cc: William Dixon; Paul Mayfield; psd@cisco.com; Zubenko, Igor;

Kartik Murthy
Subject: RE: 501 and frag comments

Excellent. One last time that I don't believe was in the spec you received. The actual vendorID payload to enable all this. I'm currently using a MD5 hash of FRAGMENTATION for the vendor id. Let me know if this is ok, or if some other string is more appealing.

It'd be nice to Interop this with you as soon as you get it running. Also, I don't think we've had full Interop testing of the basic NAT traversal stuff, either. That's even more important asap.

Not to be pushy, but any chance to review the various 501 options? We are still assessing whether 501 is necessary or not, but current word is still that we need it.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Friday, January 04, 2002 9:57 AM
To: Brian Swander; danyr@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com; Kartik Murthy
Subject: RE: 501 and frag comments

Yes, it looks like we all agree on our end.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Friday, January 04, 2002 12:54 PM
To: Rochefort, Dany; Volpe, Victor
Cc: William Dixon; Paul Mayfield; psd@cisco.com; Zubenko, Igor; Kartik Murthy
Subject: RE: 501 and frag comments

Are we agreed to this design? Any other outstanding issues?

bs

-----Original Message-----
From: Dany Rochefort [mailto:danyr@cisco.com]
Sent: Friday, January 04, 2002 9:22 AM
To: Brian Swander; vvolpe@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com;

izubenko@cisco.com; Kartik Murthy
Subject: RE: 501 and frag comments

Agreed. I was merely trying to protect the reassembly logic, but you bring up very good points that would minimize its usefulness.

-dany

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Friday, January 04, 2002 11:52 AM
To: Rochefort, Dany; Volpe, Victor
Cc: William Dixon; Paul Mayfield; psd@cisco.com; Zubenko, Igor; Kartik Murthy
Subject: RE: 501 and frag comments

This is extra complexity for little gain, IMHO. If we are fragmenting an encrypted payload (as is the common case for the ID payload), then all that can be modified is frag header fields. If they modify the encrypted data within the frag, that is identical to modifying an unfragmented, encrypted ID payload, which IKE is already robust against.

The worst this can do is cause the packet to fail reassembly and be lost, or scramble the packet during reassembly. Any attacker that can modify bits on the wire can already force IKE packets to be dropped anyway. Also, an attacker can similarily scramble the packet, encrypted or otherwise, and hashing/validation of the whole packet solves this (which we have once crypto is active).

Thus, I don't see how protecting the frag header is worth the effort. Also, then we'll have different semantics for before and after crypto keys are generated, and the problem gets much much tougher.

bs

-----Original Message-----
From: Dany Rochefort [mailto:danyr@cisco.com]
Sent: Friday, January 04, 2002 6:44 AM
To: Brian Swander; vvolpe@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com
Subject: RE: 501 and frag comments

Hi Brian,

Regarding the IKE_FRAG payload. In your proposal, you mention that someone could modify some of the data since it's not encrypted. I was wondering if you had considered HASHING the IKE_FRAG itself to allow the peer to confirm that the IKE_FRAG is intact? I realize this does create some additional overhead.

-dany

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Thursday, January 03, 2002 6:12 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com; Zubenko, Igor
Subject: RE: 501 and frag comments

No, we don't need a containing payload, since we fragment the whole IKE packet, including the IKE hdr. So, the full blow up is:

Original non-fragmented packet: IKEHDR (Encrypt, nextp=ID) Data

becomes:

IKEHDR (Noencrypt, nextp=ISAFrag) IKEHDR(Encrypt,nextp=ID), beginning of data

IKEHDR (Noencrypt, nextp=ISAFrag) more data

Etc.

Thus, instead of defining the extra fields that we care about, (nextp and hdr flags at least) in the ISA_FRAG header, I thought it much simpler to just include the original hdr, too.

This solves your concern, doesn't it? Also, since the ISA_FRAG header isn't protected, you'd have to set these fields in every frag, validate they were the same each time, etc. This duplication just didn't seem worth it.

I don't see why you'd want to change to all FFs. I know all 0's in an invalid cookie in some RFC, and all FFs is probably

valid, and I don't see what moving to all FFs would solve.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]

Sent: Thursday, January 03, 2002 1:59 PM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com; izubenko@cisco.com
Subject: RE: 501 and frag comments

Hi Brian:

One of our developers (Igor Zubenko on the CC list) has been implementing the fragmentation proposal and got it working but he ran into a few issues. One of the issues, the use of the total_frag value, we have already talked about. The other issue has to do with the ability to make the frag payload more generic. Since the contained payload value is overwritten in the IKE header, this will only work for MM pkts 5 and 6 (or for a single known first payload). Can you add a "contained_payload" field or something like that to the frag payload. This will remove the limitation and allow us to use it in different areas. For example, we are concerned that some transaction mode pkts are getting close to the MTU size. This will give us a way to get around that.

Also, Igor asked about the non-IKE marker in the port 500 version of NAT-T. His question is: why not use a cookie of all FFs. I know this was talked about early on but I do not remember the reasons why it was dismissed. Was it that all FFs is a valid cookie. If that was the only issue, NAT-T could specify that an all FF cookie has "special" meaning. Of course, there would still be the problem of IPsec aware NATs not changing the source port on IKE pkts.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]

Sent: Thursday, January 03, 2002 12:26 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com
Subject: RE: 501 and frag comments

Here's the info for the NAK:

Christian:

Don't bet on the 1/10 case. There is a quite common pattern in which fragment N is lost at every attempt. It happens for example if there is some kind of mistuned leaky bucket component if you are using IP over ATM. Please consider having at least a "NAK" of the form "please resend number N and following". Receiver should send the NAK if it receives an out of sequence packet.

bs

-----Original Message-----
From: Brian Swander
Sent: Wednesday, January 02, 2002 1:37 PM
To: Christian Huitema; Bernard Aboba; David Eitelbach; William Dixon; Ron Cully
Cc: Paul Mayfield
Subject: RE: Questions regarding the ipsec/NAT issue

Building in a NAK (or ACK) mechanism ala TCP doesn't seem essential for this release. Thus, I'd advocate not doing the partial ack for this release. Of course, IKE is extensible enough that this ACK scheme can be built in if necessary later. Indeed, if we write a draft on this, we can define an ACK scheme and make it optional to implement. If deployment experience shows it necessary, we can build it then.

Is this acceptable? I think it would be a very rare case where at lease one "fragment" is dropped on each of the retransmits. If we assume 576 MTU for IKE, a 5000 byte packet is approx 10 "fragments". We retransmit 3 times, and we'd have to have loss rates of 1/10 during each of these sends. Of course, not have an ACK scheme is bad for data transfers, but this should be ok for the IKE control traffic.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Thursday, January 03, 2002 9:15 AM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com
Subject: RE: 501 and frag comments

See below.

Victor

-----Original Message-----
From: Brian Swander
[mailto:briansw@windows.microsoft.com]
Sent: Thursday, January 03, 2002 11:59 AM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com
Subject: RE: 501 and frag comments

Answer to below:

1.Yes, I am speaking of the MTU for the interface. If you subtract the IP/UDP etc, then the MTUs for "IKE" will be smaller.

2.The example is wrong. Yes, we can do without the flags at the expense of the total_frags field in the hdr. However, that is less efficient to encode than a bit flag, and having flags around for extensibility can never hurt.

(VV) - I agree that the flags is a better way of handling this.

1.Each frag has a full IKE hdr. Thus, there will be a single reassembly per outstanding SA, which should be adequate even for a gateway, assuming that we are only fragmenting in MM, as is the case today. Of course, you can allow for multiple reassemblies per SA if you really want to. Remember, what you have is more an internal spec, not an internet draft. Thus, there are more internal implementation details than would commonly be in a draft.

(VV) - OK, that's what I thought but wanted to check.

4,5. Tradeoffs between complexity of internal state vs. messiness of the detection on the wire. We should make your proposed enhancements optional, since they won't effect interop.

(VV) - Sounds good.

Finally, (still under NDA): Christian is suggesting that we also build in a frag ack (NAK) to say: I didn't get frag #3, please resend it. This can simply be an unauthenticated notify with the data defined to say for which packet ID and fragment num, etc. we need to resend.

Personally, I believe this unnecessary complexity for the near term. He claims that some network infrastructure may consistently drop a particular packet if a stream is being sent. Do you have any experience with this? Any comments?

(VV) - I do not have any experience with this but am not sure that the ACK would solve the problem anyway. Is he saying that #3 would always get dropped if it is sent with the other fragments but would not get dropped if is sent alone. This does not sound right to me but I guess I do not know. It would be nice to not have to implement the notify. I will forward this to some people here to see if they have seen this type of a problem.

Thanks

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]

Sent: Thursday, January 03, 2002 7:40 AM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com
Subject: RE: 501 and frag comments

Hi Brian:

I read through the fragmentation proposal and overall it looks pretty good. The only thing I am not crazy about is the detection mechanism but the alternatives are worse. At least this way, only negotiations with packet loss are affected and not all negotiations. We could (and probably should) provide the user with a knob to set the IKE MTU size if the retries are taking too long or if the user knows ahead of time that there is a fragmentation issue with his NAT device.

Here are specific comments:

1. It looks like your MTU sizes are interface and not IKE based. IKE then needs to account for the MAC and IP headers accordingly so that the IKE MTU values end up being around 1480 and 550.

2. In the FRAG1 and FRAG2 example, you have a total_frags value. total_frags is not in the payload definition, should it be? If it is part of the payload, do you still need the last fragment flag?

3. When you talk about allowing a single reassembly at a time, I am assuming you mean per peer? On the gateway side we have to allow reassembly for multiple peers simultaneously. This requirement should probably be relaxed.

4. It would be nice if the responder could

learn the MTU size so that it would not have to go through the same autodetection procedure. This could be just a statement that says "If an ISA_FRAG payload is received, the receiver can assume that the IKE MTU can be no larger than the length of the received IKE packet. The receiver should update its MTU size if it is larger than the learned MTU."

5. For Continuous Channel Mode implementations, I think it would be OK to save the MTU information for rekeys. Again, this would save on the autodetection for every rekey.

I still need to go back and look at the 501 stuff.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Wednesday, January 02, 2002 5:54 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield
Subject: 501 and frag comments

Do you have any comments on this?

Still under NDA:

Apparently, we need to support 501. The current proposal, is to do to immediately float to 501 after detecting the NAT, and move the NAT-D payloads into the first payloads sent by the responder (SA payload). This is minimal effort.

The only issue I have with this approach is that it doesn't allow the peer outside the NAT to be stateless across MM rekeys. In a non-continuous channel MM approach, like we have, it is possible for all MMs to go away, and then for the external peer to need to rekey the MM. That peer will start on 500, which won't be allowed thru the NAT, and the connection will die unless the internal peer decides to rekey. This may be uncommon enough to not cause problems, but it might come back to bite us, too. That is my only concern.

The other option was:

Send an IP UDP ESP "ping" within IKE to detect the UDPESP unfriendly NATs: if it is detected, then either move to 501 to bypass the NAT, or allow the NAT to do the supposed ESP friendliness. This involves timeouts in the UDPESP unfriendly case.

bs

From: Brian Swander
Sent: Tuesday, January 08, 2002 1:42 PM

To: 'Victor Volpe'; danyr@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com; Kartik Murthy
Subject: RE: 501 and frag comments

I absolutely agree. This was designed initially with the idea of becoming the new IETF standard. Instead, it may be that the current drafts stand, and this becomes an extension. In that case, we need the actual protocol mechanism independent of the port.

To that end I suggest that we keep the current drafts as they stand. Additionally, we add:

1. extra vendor ID to SA payload, vid(501). If both sides send this, and NAT is detected in MM, then float the ID payload to start using "501". Optionally, we could do the float at the beginning of QM, too. This float would affect ALL QMs on the MM as well as any subsequent MM notifies. Opinions?
2. If you know that the peer supports 501, you can start MM on 501 at any time

How does this sound?

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Monday, January 07, 2002 10:44 AM
To: Brian Swander; danyr@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com; Kartik Murthy
Subject: RE: 501 and frag comments

OK, it does not look like the implementation will be too difficult. Just want to clear up #3. Is initiating on port 501 only acceptable on rekeys (rekeys is really the only time that state would have been saved)? If initiating on port 501 is allowed at any time, then the NAT-T payloads have to follow the same rules as they do when initiating on port 500. The difference is that the NAT-D payload end up in the KE message to avoid the compatibility problem.

Overall, this looks good to me. What is the next step?

Victor

> -----Original Message-----
> From: Brian Swander [mailto:briansw@windows.microsoft.com]
> Sent: Friday, January 04, 2002 7:10 PM
> To: Volpe, Victor; Rochefort, Dany
> Cc: William Dixon; Paul Mayfield; psd@cisco.com; Zubenko, Igor; Kartik Murthy

Subject: RE: 501 and frag comments

The proposal below is inaccurate and a little old. The plan is really to do the deterministic switch to 501 after the first 2 pkts.

```
UDP src 500, dst 500  [HDR][SA][NAT-T VENDOR ID] ---->

                      <--- [HDR][SA][NAT-T VENDOR ID][NAT-D]

UDP src 501, dst 501  [HDR][KE][NAT-D] --->

                      <--- [HDR][KE]
```

Src port of 500 I believe is an oversight in William's notes. Once you float to 501, to should float both src and dst.

Answers to below:

1. There is no delete needed since there is only 1 SA brought up.

2. Pkt loss just handled via standard retransmission scheme. Initiator will just keep sending on 501, and if no response, the connection dies

3. Yes, you should support this if you have state and know that the peer at that addr supports 501.

4. Yes, there are cornercase rekey issues. We have chosen to ignore them as likely minimal, but we can spec solutions. The scenario is:

```
C          NAT              X(external)
```

Now, we get an SA between C and X. Now, if C rekeys the MM, it is ok, since the NAT will either reuse the ports or allocate a new one. If X rekeys the MM and has state to remember to use 501, it'll be ok, since the NAT will still have a 501 mapping to allow the new request in. However, if the rare cases where X deletes all state, and then needs to rekey the MM, X would initiate on 500, and the NAT will not allow it in. Of course, the fix is to make sure that X remembers to use 501, but this is tougher to implement.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Friday, January 04, 2002 2:14 PM
To: Brian Swander; danyr@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com; Kartik Murthy

Subject: RE: 501 and frag comments

I did re-look at #1 earlier but it is really not enough to comment on. There are some things I like about it:

1. The deterministic switch to port 501 after the first 2 pkts

2. The fact that the Initiator is responsible for reinitiating from the new port

There are some things that I do not fully understand. The main issue is why a source port of 500 is still used once the switch to 501 is made?

We need to answer these questions as well:

1. Is there a delete message sent when the switch takes place or is the delete implicit?

2. What happens when there is pkt loss during the switch?

3. Can implementations accept IKE on port 501 without it starting on 500?

4. Rekey issues?

Sorry for asking for this but can you put together a more detailed description of how this is all going to work. I think a lot of this was written down long ago when you guys had your initial port 501 proposal. Do you still have the notes from that initial meeting?

I am leaving in about 10 minutes so if you do not hear back from me that is why.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Friday, January 04, 2002 4:57 PM
To: Volpe, Victor; Rochefort, Dany
Cc: William Dixon; Paul Mayfield; psd@cisco.com; Zubenko, Igor; Kartik Murthy
Subject: RE: 501 and frag comments

The best I've got is this from William earlier:

In looking at these again, and after testing NATs, both 2 and 3 assume that IPSec passthru NATs are implemented well, which is a bad assumption. We have seen that many only allow 1 connection thru in passthru mode. Thus, only #1 allows multiple connections thru any NAT.

1. Start on 500, discover the NAT, and switch to a new, non-500 port. After much hair pulling, Brian and I propose the following:

```
UDP src 500, dst 500  [HDR][SA][NAT-T VENDOR ID] ---->

                      <--- [HDR][SA][NAT-T VENDOR ID][NAT-D]

UDP src 500, dst 501  [HDR][SA][NAT-T VENDOR ID][NAT-D] --->

                      <--- [HDR][SA][NAT-T VENDOR ID][NAT-D]
```

This approach moves NAT-D to the responder in the case where NAT-T capability is detected. This avoids backward compat problems with normal IKE initiators. It adds the expense of including a NAT-D payload in every response to a NAT-T capable initiator. The new NAT-T initiator then re-initiates to the 501 port, and also includes NAT-D in the first exchange. We did this to avoid NAT-D during the KE exchange to avoid generating a DH and then discovering we have to re-initiate to move the port. This approach continues to use UDP-ESP, though with the more efficient UDP-ESP encapsulation, which will make many people happy. It doesn't increase RTs. Downside, is that it is a NAT-T draft change, and change to how anyone building ESP-UDP probably implemented already (with 0x00). Keep-alive is done for 501 only in this case to the dest IP. Firewall admins have to open 501 in addition to 501.

2. Add in MM a ping to detect the "IPSec passthrough" NAT mode.

```
UDP src 500, dst 500  [HDR:I-COOKIE=0x00:R-COOKIE=0x00][VENDORID
NAT-T] ---->

                      <---
[HDR:I-COOKIE=0x00:R-COOKIE=0x02][VENDORID NAT-T][NAT-D]
```

This stateless ping could happen as the very first packet, but then you'd always ping even before you knew there was a NAT. This might be helpful anyway if merged with the IPSec SA keep-alive somehow. And with NAT-D in the response, you can tell whether there is a NAT. If you added this ping after MM completed, then you'd know there was a NAT and you're just testing to see if UDP-ESP with 0x00 will get through. Note that the R-COOKIE must not be the same as the R-COOKIE of the outbound packet to make this test valid.

3. IKE requests a new QM IPSec SA with a normal [IP][ESP]

encapsulation, not UDP-ESP 0x00.

IKE completes MM, detects NAT, and then decides a default (or configured) way to establish the IPSec SA in QM - picking either [IP][ESP] or [IP][UDP][0x00][ESP]. If it's default, you guess which method is best for your deployment. If it's configured, it's because you know what kind of NAT you need to go through. It's possible that if 0bytes are received on the inbound SA, then you could automatically redo a QM to propose the other method. IKE implementations would have to be able to rekey QM with either proposal set after NAT is detected. Upper layer protocols that were IPsec aware, could also be aware that their initial connect was not getting through, and to request the alternate IPSec SA type.

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Friday, January 04, 2002 12:36 PM
To: Brian Swander; danyr@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com; Kartik Murthy
Subject: RE: 501 and frag comments

Brian:

I have been looking through the threads on port 501 but they are all pretty schetchy. Do you have anything that spells everything out in one doc?

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Friday, January 04, 2002 1:05 PM
To: Volpe, Victor; Rochefort, Dany
Cc: William Dixon; Paul Mayfield; psd@cisco.com; Zubenko, Igor; Kartik Murthy
Subject: RE: 501 and frag comments

Excellent. One last time that I don't believe was in the spec you received. The actual vendorID payload to enable all this. I'm currently using a MD5 hash of FRAGMENTATION for the vendor id. Let me know if this is ok, or if some other string is more appealing.

It'd be nice to Interop this with you as soon as you get it running. Also, I don't think we've had full Interop testing of the basic NAT traversal stuff, either. That's even more important asap.

Not to be pushy, but any chance to review the various 501 options? We are still assessing whether 501 is necessary or not, but current word is still that we need it.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]
Sent: Friday, January 04, 2002 9:57 AM
To: Brian Swander; danyr@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com; Kartik Murthy
Subject: RE: 501 and frag comments

Yes, it looks like we all agree on our end.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Friday, January 04, 2002 12:54 PM
To: Rochefort, Dany; Volpe, Victor
Cc: William Dixon; Paul Mayfield; psd@cisco.com; Zubenko, Igor; Kartik Murthy
Subject: RE: 501 and frag comments

Are we agreed to this design? Any other outstanding issues?

bs

-----Original Message-----
From: Dany Rochefort [mailto:danyr@cisco.com]
Sent: Friday, January 04, 2002 9:22 AM
To: Brian Swander; vvolpe@cisco.com
Cc: William Dixon; Paul Mayfield; psd@cisco.com; izubenko@cisco.com; Kartik Murthy
Subject: RE: 501 and frag comments

Agreed. I was merely trying to protect the reassembly logic, but you bring up very good points that would minimize its usefulness.

-dany

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]
Sent: Friday, January 04, 2002 11:52 AM
To: Rochefort, Dany; Volpe, Victor
Cc: William Dixon; Paul Mayfield;

psd@cisco.com; Zubenko, Igor; Kartik Murthy
Subject: RE: 501 and frag comments

This is extra complexity for little gain, IMHO. If we are fragmenting an encrypted payload (as is the common case for the ID payload), then all that can be modified is frag header fields. If they modify the encrypted data within the frag, that is identical to modifying an unfragmented, encrypted ID payload, which IKE is already robust against.

The worst this can do is cause the packet to fail reassembly and be lost, or scramble the packet during reassembly. Any attacker that can modify bits on the wire can already force IKE packets to be dropped anyway. Also, an attacker can similarily scramble the packet, encrypted or otherwise, and hashing/validation of the whole packet solves this (which we have once crypto is active).

Thus, I don't see how protecting the frag header is worth the effort. Also, then we'll have different semantics for before and after crypto keys are generated, and the problem gets much much tougher.

bs

-----Original Message-----
From: Dany Rochefort
[mailto:danyr@cisco.com]
Sent: Friday, January 04, 2002 6:44 AM
To: Brian Swander; vvolpe@cisco.com
Cc: William Dixon; Paul Mayfield;
psd@cisco.com; izubenko@cisco.com
Subject: RE: 501 and frag comments

Hi Brian,

Regarding the IKE_FRAG payload. In your proposal, you mention that someone could modify some of the data since it's not encrypted. I was wondering if you had considered HASHING the IKE_FRAG itself to allow the peer to confirm that the IKE_FRAG is intact? I realize this does create some additional overhead.

-dany

-----Original Message-----
From: Brian Swander

[mailto:briansw@windows.microsoft.com]
Sent: Thursday, January 03, 2002 6:12 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com; Zubenko, Igor
Subject: RE: 501 and frag comments

No, we don't need a containing payload, since we fragment the whole IKE packet, including the IKE hdr. So, the full blow up is:

Original non-fragmented packet: IKEHDR (Encrypt, nextp=ID) Data

becomes:

IKEHDR (Noencrypt, nextp=ISAFrag) IKEHDR(Encrypt,nextp=ID), beginning of data

IKEHDR (Noencrypt, nextp=ISAFrag) more data

Etc.

Thus, instead of defining the extra fields that we care about, (nextp and hdr flags at least) in the ISA_FRAG header, I thought it much simpler to just include the original hdr, too.

This solves your concern, doesn't it? Also, since the ISA_FRAG header isn't protected, you'd have to set these fields in every frag, validate they were the same each time, etc. This duplication just didn't seem worth it.

I don't see why you'd want to change to all FFs. I know all 0's in an invalid cookie in some RFC, and all FFs is probably valid, and I don't see what moving to all FFs would solve.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]

Sent: Thursday, January 03, 2002 1:59 PM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com; izubenko@cisco.com
Subject: RE: 501 and frag comments

Hi Brian:

One of our developers (Igor Zubenko on the CC list) has been implementing the fragmentation proposal and got it working but he ran into a few issues. One of the issues, the use of the total_frag value, we have already talked about. The other issue has to do with the ability to make the frag payload more generic. Since the contained payload value is overwritten in the IKE header, this will only work for MM pkts 5 and 6 (or for a single known first payload). Can you add a "contained_payload" field or something like that to the frag payload. This will remove the limitation and allow us to use it in different areas. For example, we are concerned that some transaction mode pkts are getting close to the MTU size. This will give us a way to get around that.

Also, Igor asked about the non-IKE marker in the port 500 version of NAT-T. His question is: why not use a cookie of all FFs. I know this was talked about early on but I do not remember the reasons why it was dismissed. Was it that all FFs is a valid cookie. If that was the only issue, NAT-T could specify that an all FF cookie has "special" meaning. Of course, there would still be the problem of IPsec aware NATs not changing the source port on IKE pkts.

Victor

-----Original Message-----
From: Brian Swander
[mailto:briansw@windows.microsoft.com]
Sent: Thursday, January 03, 2002 12:26 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com
Subject: RE: 501 and frag comments

Here's the info for the NAK:

Christian:

Don't bet on the 1/10 case. There is a quite common pattern in which fragment N is lost at every attempt. It happens for example if there is some kind of mistuned leaky bucket component if you are using IP over ATM. Please consider having at least a "NAK" of the form "please resend number N and following". Receiver should send the NAK if it receives an out of sequence packet.

bs

-----Original Message-----
From: Brian Swander
Sent: Wednesday, January 02, 2002 1:37 PM
To: Christian Huitema; Bernard Aboba; David Eitelbach; William Dixon; Ron Cully

Cc: Paul Mayfield
Subject: RE: Questions regarding the ipsec/NAT issue

Building in a NAK (or ACK) mechanism ala TCP doesn't seem essential for this release. Thus, I'd advocate not doing the partial ack for this release. Of course, IKE is extensible enough that this ACK scheme can be built in if necessary later. Indeed, if we write a draft on this, we can define an ACK scheme and make it optional to implement. If deployment experience shows it necessary, we can build it then.

Is this acceptable? I think it would be a very rare case where at lease one "fragment" is dropped on each of the retransmits. If we assume 576 MTU for IKE, a 5000 byte packet is approx 10 "fragments". We retransmit 3 times, and we'd have to have loss rates of 1/10 during each of these sends. Of course, not have an ACK scheme is bad for data transfers, but this should be ok for the IKE control traffic.

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]

Sent: Thursday, January 03, 2002 9:15 AM
To: Brian Swander
Cc: William Dixon; Paul Mayfield; danyr@cisco.com; psd@cisco.com
Subject: RE: 501 and frag comments

See below.

Victor

-----Original Message-----
From: Brian Swander [mailto:briansw@windows.microsoft.com]

Sent: Thursday, January 03, 2002 11:59 AM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield; Rochefort, Dany; psd@cisco.com
Subject: RE: 501 and frag comments

Answer to below:

1.Yes, I am speaking of the MTU for the interface. If you subtract the IP/UDP etc, then the MTUs for "IKE" will be smaller.

2.The example is wrong. Yes, we can do without the flags at the expense of the total_frags field in the hdr. However, that is less efficient to encode than a bit flag, and having flags around for extensibility can never hurt.

(VV) - I agree that the flags is a better way of handling this.

1.Each frag has a full IKE hdr. Thus, there will be a single reassembly per outstanding SA, which should be adequate even for a gateway, assuming that we are only fragmenting in MM, as is the case today. Of course, you can allow for multiple reassemblies per SA if you really want to. Remember, what you have is more an internal spec, not an internet draft. Thus, there are more internal implementation details than would commonly be in a draft.

(VV) - OK, that's what I thought but wanted to check.

4,5. Tradeoffs between complexity of internal state vs. messiness of the detection on the wire. We should make your proposed enhancements optional, since they won't effect interop.

(VV) - Sounds good.

Finally, (still under NDA): Christian is suggesting that we also build in a frag ack (NAK) to say: I didn't get frag #3, please resend it. This can simply be an unauthenticated notify with the data defined to say for which packet ID and fragment num, etc. we need to resend.

Personally, I believe this unnecessary complexity for the near term. He claims that some network infrastructure may consistently drop a particular packet if a stream is being sent. Do you have any experience with this? Any comments?

(VV) - I do not have any experience with this but am not sure that the ACK would solve the problem anyway. Is he saying that #3 would always get dropped if it is sent with the other fragments but would not get dropped if is sent alone. This does not sound right to me but I guess I do not know. It would be nice to not have to implement the notify. I will forward this to some people here to see if they have seen this type of a problem.

Thanks

bs

-----Original Message-----
From: Victor Volpe [mailto:vvolpe@cisco.com]

Sent: Thursday, January 03, 2002 7:40 AM
To: Brian Swander
Cc: William Dixon; Paul Mayfield;
danyr@cisco.com; psd@cisco.com

Subject: RE: 501 and frag comments

Hi Brian:

I read through the fragmentation proposal and overall it looks pretty good. The only thing I am not crazy about is the detection mechanism but the alternatives are worse. At least this way, only negotiations with packet loss are affected and not all negotiations. We could (and probably should) provide the user with a knob to set the IKE MTU size if the retries are taking too long or if the user knows ahead of time that there is a fragmentation issue with his NAT device.

Here are specific comments:

1. It looks like your MTU sizes are interface and not IKE based. IKE then needs to account for the MAC and IP headers accordingly so that the IKE MTU values end up being around 1480 and 550.

2. In the FRAG1 and FRAG2 example, you have a total_frags value. total_frags is not in the payload definition, should it be? If it is part of the payload, do you still need the last fragment flag?

3. When you talk about allowing a single reassembly at a time, I am assuming you mean per peer? On the gateway side we have to allow reassembly for multiple peers simultaneously. This requirement should probably be relaxed.

4. It would be nice if the responder could learn the MTU size so that it would not have to go through the same autodetection procedure. This could be just a statement that says "If an ISA_FRAG payload is received, the receiver can assume that the IKE MTU can be no larger than the length of the received IKE packet. The receiver should update its MTU size if it is larger than the learned MTU."

5. For Continuous Channel Mode implementations, I think it would be OK to save the MTU information for rekeys. Again, this would save on the autodetection for every rekey.

I still need to go back and look at the 501 stuff.

Victor

-----Original Message-----
From: Brian Swander
[mailto:briansw@windows.microsoft.com]
Sent: Wednesday, January 02, 2002 5:54 PM
To: Volpe, Victor
Cc: William Dixon; Paul Mayfield
Subject: 501 and frag comments

Do you have any comments on this?

Still under NDA:

Apparently, we need to support 501. The current proposal, is to do to immediately float to 501 after detecting the NAT, and move the NAT-D payloads into the first payloads sent by the responder (SA payload). This is minimal effort.

The only issue I have with this approach is that it doesn't allow the peer outside the NAT to be stateless across MM rekeys. In a non-continuous channel MM approach, like we have, it is possible for all MMs to go away, and then for the external peer to need to rekey the MM. That peer will start on 500, which won't be allowed thru the NAT, and the connection will die unless the internal peer decides to rekey. This may be uncommon enough to not cause problems, but it might come back to bite us, too. That is my only concern.

The other option was:

Send an IP UDP ESP "ping" within IKE to detect the UDPESP unfriendly NATs: if it is detected, then either move to 501 to bypass the NAT, or allow the NAT to do the supposed ESP friendliness. This involves timeouts in the UDPESP unfriendly case.

bs

From: Brian Swander
Sent: Wednesday, January 09, 2002 9:27 AM
To: Chris Black (NETWORKING)
Subject: RE: IPsec UDP fragmentation fix

<<ikefrag.doc>>

-----Original Message-----
From: Chris Black (NETWORKING)
Sent: Tuesday, January 08, 2002 8:42 PM
To: Brian Swander
Subject: IPsec UDP fragmentation fix

Heya Brian . . . Can you forward all the information that you have on the fragmentation fix.

Include email, docs etc. We need to make sure we can establish prior art on this.
Do you have a spec written up for this?

-- Chris

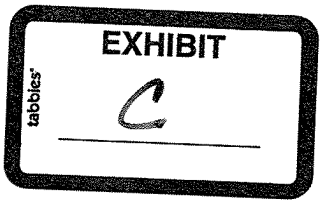


EXHIBIT
C
tabbies

IKE UDP Fragmentation support

Problem: network infrastructure, especially NATs, tend to drop IP fragments. The IKE ID payload is often fragmented when it carries large cert chains.

Solution: incorporate limited fragmentation/reassembly and MTU discoverability via black hole detection into IKE.

Big picture:

Send big ID payload as normal. Wait for response. Retransmit once or twice to allow for normal lost packets, and slow peer validation. Then, begin black hole detection. Set MTU at 1500, and send packet in 1500 byte fragments. If still no response, set MTU to smallest dialup MTU (576), and resend fragments.

Implementation: ID payload is constructed and encrypted as normal. In sending routine, we will check if we are doing fragmentation. This can only be done if the peer is a MSFT (or compatible) implementation of sufficient version (as determined by the exchange of the vendor IDs earlier). If we are doing fragmentation, we take the normal payload:

IP UDP IKEHDR [ID CERT SIG] where [] denoted encrypted

And send:

IP UDP IKEHDR FRAG1
IP UDP IKEHDR FRAG2

The FRAG is a new IKE payload type ISA_FRAG:

```
typedef struct frag_payload_ {
    unsigned char next_payload;
    unsigned char reserved;
    unsigned short payload_len;
    unsigned short fragment_id;
    unsigned char  fragment_num;
    unsigned char  flags;
} frag_payload;
```

FRAG1 above will have {id=1, num=1, total_frags = 2};
FRAG2 will be (id=1,num=2, total_frags=2, flags = LAST_FRAGMENT};

```
#define FRAG_FLAG_LAST_FRAGMENT 0x1
```

This is considerably simpler than IP since we don't need to worry about more fragmentation happening within the network itself.

To avoid attacks and buffer management issues, we will only allow one outstanding reassembly at a time. As soon as a packet with a new fragment ID arrives, it is a new fragment, and all outstanding state from a potential old reassembly is discarded. This is acceptable since IKE in MM is a lockstep protocol. If QM messages start needing to be fragmented in the future, then this requirement will need to be relaxed.

In addition, we keep track of the total number bytes buffered for this reassembly. When that exceeds a maximum (currently 64k), we discard the buffers.

This is currently coded to be as safe as possible. This means that I have traded off optimized buffer management for safety. Also, I reassemble the entire packet, and then inject as if it were received from the wire, so there are no new code paths to test.

Retransmissions:
Initial send - full packet
Retrans 1, 1 second later: full packet
Retrans 2, +2 seconds, 1500 MTU fragments, if peer correct version
Retrans 3 and beyond, 576 MTU fragments

Security issues:

The IKE Frag header will be unencrypted, even if the rest of the payload is encrypted. This means that people can mess with all the fields in the frag header. The most this can cause is scrambling or dropping of the packet. This can always be caused by man-in-the-middle packet modifications today as well.

Buffer attacks and reassembly attacks (ping of death) are obviated by the simple reassembly options, and only keeping one reassembly state outstanding at a time.

Another issue to be careful is handling chained payloads. Since the FRAG payload is unencrypted, we need to special case processing when we normally expect encrypted data, as is the case when sending the ID payload. Thus, we need to make sure that an attacker cannot send a small FRAG payload, and then chain in other stuff that we'd process in the clear (i.e. without verifying the encryption first). This is done in a few steps:

1. any next payload after a FRAG is ignored
2. only if the first payload is a FRAG (correct case) is encryption ignored
3. Non-MSFT implementation FRAG payloads are discarded

Timeframe:

I estimate 1 week to dev, and a couple of days to unit test. The destabilization isn't too scary, but this will still need a fair amount of stress and scenario coverage.

Testing:
I'm done with unit testing. This includes small scale stress testing to make sure there are no leaks. In addition, I verified that the packet is deleted correctly if all fragments are not properly received, but the next resend is ok. Finally, I made sure that the out of order fragments are correctly reassembled.

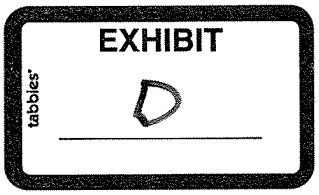


EXHIBIT
tabbies
D

Disclosure Packet

MS#: 300024.01
Title: Packet Fragmentation
Microsoft Team: Dan Christen, Joe Hoggard, Noemi Tovar
Inventors: Brian Swander, Ron Cully

Dev Group: Networking-IPSec

Summary: The invention addresses the problem of dropped IP packets, especially Internet Key Exchange (IKE) ID payloads, by incorporating limited fragmentation/reassembly and Maximum Transmission Unit (MTU) discoverability via black hole detection into IKE.

Initially, large ID payloads are sent in a normal fashion, with the sender waiting for a response. If there is no response, the sender retransmits once or twice to allow for normal lost packets, and slow peer validation. If there is still no success, then the black hole detection scheme is started. Black hole detection involves setting the MTU to progressively smaller fragment sizes to see if a particular size is able to get through.

This is networking invention relates to NATs (Network Address Translators), which often cause dropped IP packets. NATs are currently an active area of research/development and the subject of a number of proposed standards.

Technology: Communications Protocols and Data Formats, Networks, Security and Authentication.

File by Date: 1/18/2002

Underlying Facts:

- RUSH! We have scheduled time with the inventors for 1/14/2002 when they are all available and want to VTC with OC during this time if available. This area is hotly contested and it would be advantageous to file by end of next week.
- Date of Conception – Dec. 5, 2001
- Reduced to Practice? – Yes

- Publicly Disclosed? – No

Foreign Filing Intentions: (Y/N) Yes, subject to budget limitations

Publication Intentions: (Y/N) Yes

Prior/Related Art: (including similar MS apps) None

Attachments: (disclosure and other supporting docs)


ikefrag.doc


fragmail.txt

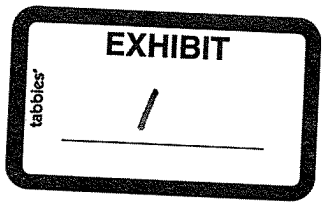


EXHIBIT
/
tabbies

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Swander, et al.	Examiner:	Jeffery L. Williams
Serial No.:	10/056,889	Group Art Unit:	2137
Filed:	January 25, 2002	Docket No.:	14917.0431US01
Title:	METHOD AND APPARATUS FOR FRAGMENTING AND REASSEMBLING INTERNET KEY EXCHANGE PACKETS		

DECLARATION OF BRIAN SWANDER & CHRISTIAN HUITEMA

PURSUANT TO 37 CFR 1.132

We, Brian Swander and Christian Huitema, hereby declare as follows:

1. We are joint inventors named on U.S. Patent Application Serial No. 10/056,889, filed January 25, 2002 (hereinafter, "the present application").

2. We are co-inventors of the subject matter disclosed and claimed in the present application.

3. We are aware of the Office Action in the present application mailed December 8, 2006 in which Examiner Jeffery L. Williams maintained "Minutes of IPSEC Working Group Meeting" (hereinafter, "IPSEC Minutes") in view of "Fragmentation Considered Harmful" to Kent as a basis for rejecting independent claim 1 of the present application under 35 USC § 103(a).

4. We are aware of an Office Action Response and an Amendment in the present application being filed in response to the Office Action and that this declaration is attached to the Amendment as part of Exhibit 1.

5. We are aware that any material derived from us may not be used as prior art against our invention and patenting thereof, unless that material was publicly disclosed more than a year before our filing of the patent application. We are also aware that to show that material in the prior art derived from us we will need to state unequivocally as such and provide objective evidence as to the same. We hereby aver that the subject matter provided in the "Minutes of IPSEC Working Group Meeting" reference derived from us.

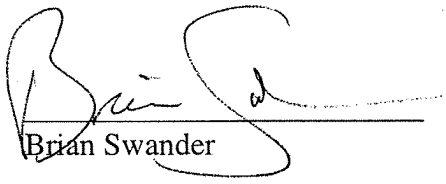
6. We do not believe that the IPSEC Minutes describe our solution to IKE fragmentation as embodied in the claims of the present application. However, even assuming that the IPSEC

Minutes somehow were construed to disclose the subject matter, the disclosure in the IPSEC Minutes derived from us.

7. Examiner Williams has cited a portion from the IPSEC Minutes attributed to William Dixon. (Exhibit 2 at Page 2). The presentation given by William Dixon at the IPSEC meeting is attached as Exhibit 3. At the time of the IPSEC Minutes recordation, William Dixon was an employee of Microsoft and worked closely with us. In out interactions at Microsoft, we provided to Mr. Dixon, in one or more conversations, all the information regarding IKE fragmentation presented by William Dixon in the IPSEC Minutes. William Dixon attests to these conversations and our disclosure to him, as evidenced by his affidavit attached as Exhibit 4. Therefore, all disclosure cited by Examiner Williams in the IPSEC Minutes derived from us.

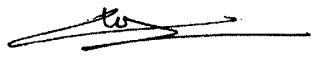
8. We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that statements are made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such false statements may jeopardize the validity of the application or any patent issued thereon.

Date 3/1/2007



Brian Swander

Date 3/1/2007



Christian Huitema

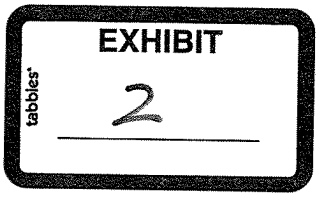


EXHIBIT
tabbies
2

Current Meeting Report

Minutes of IPSEC wg Meeting, December 2001

These minutes were based off of notes graciously taken and submitted by David Black, from EMC.

Wednesday, December 12, 2001 (15:30 -- 17:30)

Agenda Bashing (Ts'o)

The following agenda was presented and discussed:

Wednesday, December 12, 2001 (15:30 -- 17:30)

* Agenda Bashing (Ts'o)
* SCTP/IPsec draft (Angelos)
* IPSEC over NAT (Dixon)
* IP Storage Security (Aboba)
* AES Cipher document(s) (Herbert / Frankel)
* Revised ESP (Kent)
* IPSEC performance (Angelos)
* Suggested Identity draft (Sommerfeld)
* Transport Mode for Virtual Networks (Touch)
* IKE Implementation Issues (Richardson)
* Son-Of-Ike Requirements (Madson)

Thursday, December 13, 2001 (9:00 -- 11:30)

* JFK proposal (Angelos / Bellovin)
* IKE V2 proposal (Radia / Dan Harkins)
* IKE-SIGMA (Hugo / Bitan)
* Requirements and Comparison of SOI approaches (Kaufman)
* SOI Performance comparison (Rescorla)
* Son-Of-Ike Requirements (Madson)
* Open Mike Period

SCTP/IPsec draft (Angelos Keromytis, Columbia University)
==========

Slides available in postscript.

Angelos gave a description of the issues raised in WG last call. The main issue was to rename ID_RECURSE to ID_LIST for clarity. A new draft will be issued to address this and a few other minor issues, and the document will then be sent to the IESG.

IPSEC over NAT (William Dixon, Microsoft)

The NAT Traversal documents are in last call. There was recently implementation testing among 8 vendors against various NAT devices. What was tested was L2TP in IPsec transport mode and in tunnel mode based on IKE extensions of IKE-CFG/XAUTH or IPSRA DHCP.

The results from this testing revealed some problems. Certificate fragmentation was a major cause of problems. In addition, some devices seem to be looking at IKE cookie states and hence cause NAT traversal not to work.

A list of possible approaches to avoid fragmentation was discussed. Some testers implemented fragmentation avoidance via multiple UDP packets (fragment above UDP to avoid IP fragmenting below UDP). Something will likely need to be done here, since fragmentation will be a real deployment obstacle.

IP Storage Security (Bernard Aboba, Microsoft)
==========

Slides available in Powerpoint format.

IP Storage WG has adopted IKE and IPsec as basic security mechanisms. The IPS working group has dependencies on various IPSEC and IPSRA wg I-D's, including IPsec transforms (3DES, HMAC-SHA1, AES-CTR, and AES CBC MAC w/XCBC), and tunnel mode config/auth (draft-ietf-ipsec-dhcp-13.txt - being done in IPSRA WG)

A number of issues are under discussion. Please read draft-ietf-ips-security-06.txt and send comments to the IPS WG mailing list (ips@ece.cmu.edu) or the draft authors (iscsi-security@external.cisco.com).

AES Cipher document(s) (Sheila Frankel, NIST and Howard Herbert, Intel)

Three AES Cipher documents were discussed (one encryption, and two MAC algorithms):

1. AES cipher: CBC with random IV. 128 bit key is mandatory, 192 and 256 are optional, key length attribute is mandatory in IKE Phase 1 and Phase 2. Have suggested DH groups for various AES key lengths, and authors will resolve this with other different key length recommendations (e.g., Hilary Orman's key length comparison draft).
2. HMAC-SHA-256: Motivation is to increase the key space so that rekeying frequency can be reduced. The hash is truncated to 96 bits to avoid packet size changes.
3. AES-XCBC-MAC-96: Problem is that CBC MAC isn't secure for variable length messages. XCBC corrects this and has no patent issues --- do not confuse this with a new proposed "combined" AES confidentiality+integrity mode that does have serious IP issues. Comments from one of the original XCBC authors will be incorporated, test vectors added, and next version of draft is due in January 2002.

Revised ESP (Steve Kent, BBN/Verizon)

==========

Steve presented some proposed changes to ESP. The changes had a few clarifications (such as using "integrity" instead of "authentication"), and a number of substantive changes:

* Sequence number extension - This is needed to support higher speed systems, and anticipates AES and its longer lived connections. New sequence numbers are 64 bits, but only 32 bits are carried in each packet. The next version of the draft will have in its appendix a suggestion on how to deal with the rare case where 2^32 packets from one SA are dropped.
* Support for combined modes - New algorithms have emerged that can do integrity and confidentiality in one pass. This requires slight changes to ESP packet format - algorithm has to specify how it checks integrity, as this can't be specified once for all possible combined modes.
* Improved traffic flow confidentiality - only for tunnel mode, most effective between a pair of security gateways. Have expanded the allowed padding (put "junk" behind the IP packet, and encapsulated IP header's length field will automatically cause receiver to ignore the "junk"). Also suggest a convention that the next header field of "59" in the ESP frame be used to indicate that there is no IP packet --- this allows dummy packets to be sent and discarded quickly to obfuscate traffic analysis.

There were some discussions about fragmentation, and whether we can "just say no" to fragments. Steve Kent observed that fragmentation in IPsec has disastrous performance implications, and causes receiver administration issues because port selectors can't be applied to fragments (have to reassemble), which in turn leads to a denial of service vulnerability based on consumption of reassembly resources (e.g., buffers). Also see NAT discussion about other problems caused by fragments.

Several problems with simply getting rid of fragments were raised as the discussion continued:

* Some systems still don't do Path MTU discovery and this is made worse by firewalls that discard ICMP and hence break Path MTU discovery.
* IKE/NAT problem only affects IKE messages that carry certificates. That's a much smaller scope than ESP in general.
* We have no choice but to deal with fragments on reception, as one can't be sure sender send DF or some intermediate system paid attention to it. Not sending fragments is a good idea.
* Some firewalls don't like either fragmentation or ICMP, making the situation impossible as fragments get dropped and Path MTU discovery doesn't work. (Steve Kent observed that if the firewall is broken regardless, we should do the simplest thing in ESP.

Steve then discussed other document revisions which he is planning to do:

* AH - revision early next year based on ESP revisions (e.g., extended sequence numbers)
* Architecture document - revision before Yokohama to simplify processing model, reduce requirements for nested SAs, remove MUST for AH, selector changes (e.g., along lines SCTP needs, plus allow ICMP selectors).

There was a discussion about whether or not the distinction between tunnel and transport modes

could be eliminated. Steve Kent responded that the problem is making sure that the right selector checks happen; Joe Touch's VPN draft is a good model to follow. Steve rejected a proposal to remove the tunneling specification from IPSEC, and to simply reference some other specification (such as IP-IP tunnel), on the grounds that encapsulation/decapsulation decision on whether or not to propagate fields have security consequences, and thus need to be made based on local security policy decisions.

IPSEC performance (Angelos Keromytis, Columbia University)
==========

Slides available in postscript format.

Angelos gave a presentation which measured the performance of IPSEC using DES, 3DES, and AES in various scenarios, using both hardware and software implementations.

Suggested Identity draft (Bill Sommerfeld, Sun)
==========

Bill Sommerfeld discussed an individual I-D submission (draft-keromytis-ike-id-00.txt) which adds a suggested identity field to the IKE negotiation. This addresses the problem of how to figure out which IKE identity to use when there's more than one. The field is added to initiator's message 5 and HASH_I. Allows initiator to suggest which id responder should use to avoid confusion. This is needed for for User-to-User keying and Responder-initiated rekeying.

Bill would like this to be adopted as a WG draft. He requested that the working group read the I-D and comment on mailing list.

Transport Mode for Virtual Networks (Joe Touch, USC/ISI)
==========

Slides are available in Powerpoint and PDF format.

Joe gave a presentation on issues relating to using tunnel-mode IPSEC and hop-by-hop routing. This causes complications because you either need to violate layering one way or another (for example, the routing layer has to update IPSEC configuration as the routing changes).

Joe presented a solution which uses IP-IP encapsulation (RFC 2003) and IPSEC transport mode. The result is syntactically identical to IPSEC tunnel mode, although security checks which are done upon receipt and decapsulation of the packet are different.

Joe then asked the question of how the working group should handle draft-touch-ipsec-vpn-*.txt. Should it become an informational RFC, or a BCP? He also requested that the next revision of RFC2401 require transport mode in gateways and allow the approach which he outlined. He also requested that in the son-of-ike proposals, that tunnel configuration be separated from keying.

There was then a discussion about the order in which key selection and forwarding should be

done. The resolution is that having forwarding select a virtual interface and using SPD per virtual interface is allowable by current documents (this is what Joe wants), but the current 2401 text could be clarified.

IKE Implementation Issues (Michael Richardson)
==========

Slides available in Postscript format.

Michael Richardson discussed draft-spencer-ike-implementation-00.txt, which documents a number of implementation issues noted by the Free S/WAN developers. The first major issue is whether "unique" IKE message Id's have to be truly unique, or whether they just need to be generated in a pseudo-random fashion, and simply "probably unique". Many implementations do the latter, and the RFC's are ambiguous on this point. Michael would like the RFC's to be changed to make it clear that implementations must keep track of every message id ever issued by an implementation to guarantee uniqueness.

The second issue related to how rekeying phase 2 SA's should be handled, and Michael proposed a scheme where the new Phase 2 SA is starts getting used immediately for transmission as it is negotiated, but the old SA is kept for in-flight messages.

The Draft also contains a bunch of other things about IKE (e.g., pieces of IKE that Free S/WAN didn't implement, and whose absence has not caused interoperability issues).

Son-Of-Ike Requirements (Cheryl Madson, Cisco)
==========

Cheryl Madson discussed her I-D, draft-ietf-ipsec-son-of-ike-requirements-00.txt. The goals of this document were to describe the characteristics of an optimal protocol, and to provide scoping by describing the scenarios which the protocol must be able to accommodate. Explicit non-goals were (a) discussing security requirements (which is tough to do in a fashion which is meaningful, but which doesn't favor one proposal or another), and (b) determining the exact split of responsibility between Son-of-Ike and other protocols for the entire set of things that are needed to set up a secure connection.

Cheryl listed the following desirable characteristics:

1. Extensibility. Can't add payloads to IKEv1, as it results in failure of negotiation. Vendor-ID has been used to work around this, but it's not a good solution.
2. Modularity.
3. Improved Convergence. It's too easy for IKEv1 participants to get out of sync with each other (e.g., SA deletion, error conditions)
4. Simplicity, both in terms of overall protocol functionality extent, and ease of accomplishing a particular function.
5. Better discussion/specification of authentication to deal with "IKE requires X.509" myth and remove sources of interoperability problems.

A proposed way of accommodating this would be to allow authentication to be plugged in via companion drafts, while keeping the base draft as simple as possible.

Document currently contains the following scenarios (list is incomplete):

* Site to site VPN
* Remote access
* End to end
* Mobile IP

The working group is asked to review the document, and propose additional scenarios as appropriate.

Thursday, December 13, 2001 (9:00 -- 11:30)
==========

JFK proposal (Angelos / Bellovin)
==========

JFK Design Process - Steve Bellovin, AT&T Labs - Research

Slides available in Postscript and PDF format.

Steve Bellovin started by describing the design principles that were used in writing JFK.

The JFK team decided that patching code to preserve IKE is the wrong thing to do - IKE is already too complex, and complexity leads to security bugs. Wanted provable correctness, DoS mitigation, no negotiation. Orthogonal design and clean, well-defined interfaces to cryptographic core. IKEv2 authors have done a great job in simplifying IKE, but the result is still too complex.

Current draft documents only the cryptographic core, if WG views this direction as valuable, a complete version of the draft will be available for Minneapolis.

JFK currently requires certificates. IKE's multiple modes were also removed. Current pre-shared (secret) key approaches are to be replaced with self-signed certificates or IPSRA certificate retrieval. For legacy authentication, IPSRA or KINK should be used.

Phase 1 vs. phase 2 distinction was also removed. This was motivated in part by DES weakness that required frequent rekeying - AES does not have this problem.

The JFK design team is prepared to modify JFK to match developing state of requirements, since the requirements draft is still a work-in-progress.

JFK Protocol - Angelos Keromytis, Columbia University

Slides available in Postscript format.

Angelos described the actual details of the JFK protocol. In essence, JFK uses a two round trip protocol. The responder keeps no state between receipt of messages 1 and 3 to avoid denial of service attacks. More details are present in the slides and in JFK I-D. Angelos noted that the draft describes keying a unidirectional SA, but it would be straightforward to key a pair of keys (one for each direction) as IKEv2 does.

One notational error was noted in the slides; there were four exponentials used (g^x, g^y, g^i, and g^r), but there should have been only two exponentials in use. (g^x and g^y should have been g^i and g^j, respectively).

In JFK, the Responder exposes her identity in Message 2, but the Initiator's identity is protected. To protect responder's identity, pick up Hugo Krawczyk and Ran Cannetti's suggestion to incorporate SIGMA ideas into JFK - this protocol is being called LBJ.

4 implementations of JFK are being done by students at Columbia - 2 in Java, 1 in C, 1 in Perl. The Java implementations are interoperating, C and Perl aren't quite there yet. About 3 weeks of student effort so far. C and Perl implementations are running into crypto library issues (e.g., padding is different in different libraries). Converting a JFK implementation to LBJ took a student about a day.

The sizes of the messages are at most a few hundred bytes plus the certificates. Comment: JFK has imperfect forward secrecy. Same D-H exponent used across multiple exchanges with forward secrecy established once that exponent is replaced. Perfect forward secrecy would require state to be kept after receipt of message 1.

Comment: Can IKE payload formats be used? Yes, message format in JFK draft was chosen for expediency and is simpler than IKE's, but can use IKE's payload formats.

Comment: Phase 2 can be useful for more than rekeying, and ability to amortize cost of public key operations is valuable. This needs to be taken up in requirements discussion.

Proof of security is in the works - not completely done yet.

IKE V2 proposal (Radia / Dan Harkins)
==========

Slides available in Postscript format.

Most of the work in IKEv2 was in the rest of the stuff that surrounds the cryptographic exchange. IKEv2 also has a 4-message exchange, but this other stuff is at least as important.

The goals of IKEv2 were listed as follows:

* Consolidate RFCs 2407, 2408, and 2409 into a single document.
* No gratuitous changes, but simplify as appropriate (e.g., phase 2 has been kept, now 1 possible phase 1 exchange as opposed to 8 in IKEv1).
* Fix ambiguities and bugs.
* Add flexibility where necessary (e.g., selectors)
* Reduce latency by reducing message count.
* Allow stateless cookies

IKE SA + IPsec SA in established in 4 messages based on public signature keys. Both identities are hidden from passive attackers. Subsequent child SA's require two messages to set up. All messages are request/response, and messages have sequence numbers. Multiple concurrent requests can be issued in parallel.

Version numbers and critical flag defined to enable future changes and extensions, to provide for forward compatibility.

Traffic selectors have been generalized. Responder can narrow ranges.

Cookies (initiator-responder pair) are used to identify IKE SAs.

In IKEv2 SA lifetimes are NOT negotiated. Either side can rekey at any time, and rekeying the IKE SA inherits all of the child SA's. No dangling SA's are allowed. If an unauthenticated ICMP/IKE message raises a suspicion about a dead peer, this is checked by sending a reliable IKE message; if there is no response, the SA is deleted.

The way IKEV2 messages are encrypted and integrity protected is done using the ESP format, but this was simply a reuse of the syntax, and not the protocol. This caused some confusion because some people thought this resulted in a bootstrapping problem, where as it was merely the intention of the document authors to be lazy and not need to reinvent the wheel. The next version of the IKEv2 draft will copy the ESP syntax by value instead of by reference to eliminate this confusion.

IKEv1 has a problem with security parameter negotiation in that each additional parameter to be negotiated results in an exponential explosion of possibilities. To address this, IKEv2 uses a "chinese menu" approach --- i.e., any of these encryption transforms with any of these integrity transforms.

The following comments were made by members of the working group:

* Rekeying IKE SA should use PFS.
* Critical bit on options has been abused to defeat interoperability in other protocols

IKE-SIGMA (Sara Bitan, Technion)
==========

Sara Bitan gave this presentation since Hugo was not able to attend the IETF meeting.

The focus of IKE-SIGMA is crypto design, similar to JFK. Get this right, then add the rest of the structure, which is orthogonal to the core key exchange crypto protocol.

SIGMA supplies full or windowed PFS, identity protection (against active or passive attacks) and DoS resistance. The protocol is flexible to allow choices in these areas.

The presentation was a walk through of the cryptographic thinking that leads to the design of the SIGMA exchanges. Strong binding of identities of the participants to the key exchange is an important theme - leads to a requirement for each sender to MAC its own identity in the protocol. Several variants of SIGMA protocol based on different choices of security properties/features were described.

Requirements and Comparison of SOI approaches (Kaufman)
==========

Slides available in Powerpoint format.

Charlie Kaufman gave a presentation which compared the security properties of the various SOI approaches, as described in the I-D at the time of the meeting.

The differences between the various protocols' security properties can be broken into a number of different areas:

Performance - number of messages, number of exponentiations, size of messages, amount of data that needs processing. There were no major performance difference in computational time among proposed new protocols.

Stateless cookie - defense against computational denial of service attack. This is easy to do with two extra messages. IKEv1 doesn't support this. JFK can piggyback this (no extra messages). SIGMA puts in two extra messages when under attack. IKEv2 can do either.

Identity hiding - again, easy to do with extra messages. Discussion of identity hiding properties, consequences. JFK exposes Bob's identity, no other protocol exposes either identity. JFK and SIGMA as published are subject to polling attacks (poll IP address, discover Bob is there), but IKEv1 and IKEv2 allow Alice to be tricked into revealing her identity - this is a "no-free-lunch" tradeoff as one or the other has to be possible based on who reveals their identity first.

Dead Peer detection - relying on ICMP opens up a denial of service attack based on forging ICMP messages. IKEv2 bans dangling SAs and relies on ping over IKE SA to avoid this. Other protocols don't say anything, but this is also not a problem in practice (yet). Putting ping into ESP and AH may be an alternative.

Plausible Deniability. JFK - can prove that the two named parties intentionally talked to each other. Others can prove that each named party talked to someone. Use of pre-shared key (or IKE

encryption key) can make it impossible to ever prove anything.

Parameter Negotiation - seems to be necessary for various reasons. Need to avoid active attack that results in weakened crypto. JFK has Bob choose IKE parameters, will add ESP/AH negotiation later. IKEv2 has same abilities as IKEv1, SIGMA continues IKEv1 approach.

IKEv2 intends to replace RFC 2407, 2408, and 2409. Other two currently supplement them, but JFK intends to get to same level of completeness as IKEv2

SOI Performance comparison (Eric Rescorla)
==========

Eric Rescorla gave a presentation which counted the number of messages and cryptographic operations (i.e., D-H key agreement, RSA private key operations, etc.) to give a comparison of the various SOI proposals.

Son-Of-Ike Requirements (Cheryl Madson, Cisco)
==========

After the presentation of the SOI proposals, Cheryl Madson returned to lead a continued discussion on the requirements draft. One of the key areas which still needs more effort in the requirements draft is the scenarios description to provide protocol scoping. The individual scenarios need refinement; in addition a model to evaluate the scenario is still needed.

There is also a need to figure out where the homes are for things that IKE will not do that are still important and get those specs done in the same timeframe as Son-of-IKE. If some of this is involved in connection establishment, the state machine specification should accommodate this.

In general, we want minimal message size and count, minimal processing expense (including light-weight devices and restart hit after crash on a large server).

Open Mike Period
==========

Jeff Wong (Cisco): Limit amount of policy provisioning in IKE to stuff absolutely required to set up IPsec SA and tunnel (if used). Policy provisioning in general is arbitrarily complex.

Hilarie Orman (Novell/Volera): Move all policy stuff into a separate protocol, e.g., see the IPSRA work.

William Dixon (Microsoft): Scenarios are the most important piece of the requirement discussion.

Cheryl: Want all the help that I can get in this area.

Michael Thomas (Cisco): Most IKE problems are in the mundane protocol operations stuff, not

the crypto itself. Recovery from restart avalanches is an important issue in practice today - KINK is in better shape on this than IKE, and needs to be looked to for examples.

Phil Hallam-Baker (Verisign): XKASS is a serious proposal in XML world - may or may not be appropriate for IPsec based on requirements. Need to think about whether credential needs to include identity information, as identity hiding is easy if there's no identity to hide.

Paul Hoffman (VPN Consortium): Transition from and coexistence with IKEv1 in a non-confusing fashion is important.

Steve Bellovin (AT&T Labs Research): Strongly agree that the non-crypto stuff needs attention (Steve is one of the JFK folks).

Cathy Meadows (??): Wants clarity on JFK vs. IKEv2 requirements/philosophies/approaches to SA negotiation.

Steve Bellovin: Can WG please tell us what the requirements are? Absent that, JFK will pursue an approach of "here's what I want" responded to with "Yes" or "No and here's why". Aim is to reduce options/negotiations.

Radia Perlman: IKEv2 follows general approach of IKEv1 with compaction and simplification. Could also follow TLS approach of offering entire suites as opposed to independently negotiating components. Locking down to a single crypto suite seems to be too restrictive.

William Dixon (Microsoft): A real scenario for identity hiding is that Microsoft would not like to reveal that a home PC is owned by MS and used by an MS employee, as hackers are specifically targeting those.

Eric Rescorla (??): Have seen four key exchange protocols and four variants, and they are similar at a high level (e.g., number of round trips). What if we started with the protocol infrastructure and then plugged in the crypto (any of these proposals seems workable)?

Uri Blumenthal (Lucent Bell Labs): Legacy authentication will live on for a long time - please do not limit new algorithm to public key.

Markku Saavla (sp? Siemens): Can we identify scenarios that are out-of-scope in addition to scenarios that we do need to support?

Cheryl: Something like that - scenarios in requirements draft were intended as the "absolutely crucial to support" ones.

Charlie Kaufman (IBM): IKEv2 copied IKEv1 syntax to avoid changing things. JFK picked up a new, cleaner syntax, but this issue is fundamentally a matter of taste. How can we avoid spending too much time in this sort of rathole? Suggests that negotiation approach needs to be resolved early (e.g., reduce everything to a small number of crypto suites, and force things like "vanity crypto" to use vendor payloads).

Cheryl: Use of suites simplifies protocol analysis.

???: Managed IP service provider is concerned about credential management and delivery - need a pre-shared mode of some form, as this is easy to manage. Please consider credential delivery and management as part of Son-of-IKE.

John Ionnadis (??): Split requirements into three categories -
Cryptographic, Operational, and Policy (e.g., negotiation). Pursue these more or less independently and in parallel.

Barbara Fraser (Cisco, co-chair): Yes, something like that is important to structuring the discussion and draft (or drafts). Scenario discussion is also important to this.

Ted Ts'o (IBM, co-chair): Yes, discuss these independently on mailing list and close them independently.

Dan Harkins (??): Would like to delete some of the proposal parsing stuff from IKE (AH and/or ESP and/or IPcomp).

Kathy Meadows (??): Supports this, suggests formal language for expressing proposals.

Jeff Schiller (Security AD): If WG spends too long debating proposals, AD will consider shutting it down. This area is reasonably well-understood and hence coming up with a reasonable solution should happen in short order. WG chairs will be asked to come up with a schedule, and hold WG to it.

Slides

The AES-XCBC-MAC-96 Algorithm and Its Use with IPsec
Overview of IKEv2
Some IPsec Performance Indications
Just Fast Keying (JFK)
SCTP and IPsec
'draft-spencer-ipsec-ike-implementation-01'
AES-Related Drafts
Engineering Trade-offs in Authentication Protocols
IPSec over NAT Testing
ESP v2- What's New?
IPsec Transport Mode for Virtual Networks
Requirements Discussion
Son of IKE Protocol Reqts
SIGMA: SIGN-and-MAC
Overview of IKEv2
Some IPsec Performance Indications
Just Fast Keying (JFK)

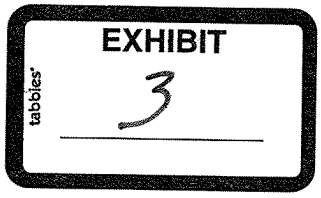
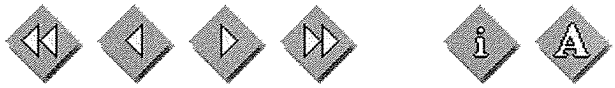


EXHIBIT
tabbies'
3

IPSec over NAT Testing

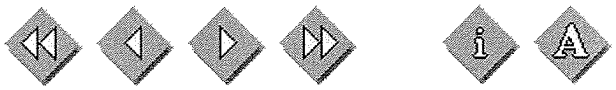
William Dixon
Windows Operating System Division
Microsoft
52nd IETF IPSec WG Dec 12 2001



Slide 1 of 7

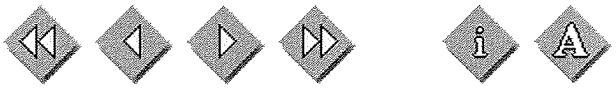
IPSec VPN Remote Access Packet Formats

- L2TP protected by IPSec transport mode
- IPsec tunnel mode using IKE extensions of IKE-CFG/XAUTH, or IPSRA DHCP



Slide 2 of 7

PKI Hierarchy Deployment



Slide 3 of 7

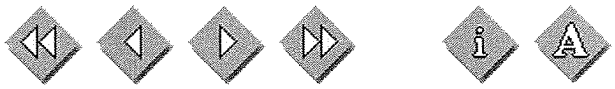
L2TP/IPsec with UDP-ESP NAT Results

NAT device	Certs w/frag	UDP 500 With PSK	UDP 500 Nofrags	UDP 501 w/frag
Win2k/XP ICS	Pass	Pass		Pass
Watchguard SOHO	Fail	Fail		
Linksys - BEFSR41	Fail	**Pass**	**Pass**	
SMC Barricade	Fail	Fail	Fail	
NetGear - RT311	Pass	Pass	Pass	pass

Slide 4 of 7

L2TP/IPsec with UDP-ESP NAT Results

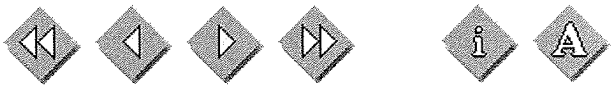
NAT device	Certs w/frag	UDP 500 With PSK	UDP 500 Nofrags	UDP 501 w/frag
Buffalo - WLAR-L11-L	Fail	**Pass**	**Pass**	**fails w/o frag avoidanc**
2Wire - Home Portal 100W	Fail	Fail		
Compaq Ipaq Connection	Fail	**Pass**	**Pass**	**fail w/o frag avoidance**



Slide 5 of 7

Conclusion

- UDP-ESP is ok, just a few devices with "IPSec passthrough" via IKE cookie aware
  - Incomplete set of devices in test matrix ?
  - Device NAT soft/firmware can be field upgraded ?
- Fragmentation of IKE ID exchange is now the biggest deployment blocker



Slide 6 of 7

Potential Frag Solutions

- **Reduce the problem**
  - How small is required for UDP packets ?
    - Internet DNS Root Operators studied problem – 576 IP is required
  - Smaller IKE payloads
    - Don't send PKCS#7 chain, only end-entity cert
    - Doesn't solve Internet MTU
  - Smaller ESP payloads
- **Fix the NET**
  - Fix NAT devices to re-assemble or track frags
  - Fix router filtering devices to track fragments
- **Fix IKEv1**
  - IKE vendorid
  - Failover to TCP
- **Provide solution in IKEv2**

Slide 7 of 7

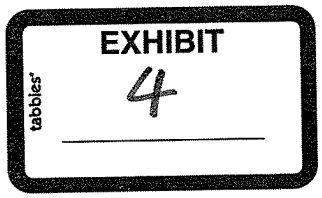


EXHIBIT
4
tabbies

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Swander, et al.	Examiner:	Jeffery L. Williams
Serial No.:	10/056,889	Group Art Unit:	2137
Filed:	January 25, 2002	Docket No.:	14917.0431US01
Title:	METHOD AND APPARATUS FOR FRAGMENTING AND REASSEMBLING INTERNET KEY EXCHANGE PACKETS		

DECLARATION OF WILLIAM DIXON

PURSUANT TO 37 CFR 1.132

I, William Dixon, hereby declare as follows:

1. I am the author of the IPSEC over NAT Testing presentation given in the December 12, 2001 IPSEC Working Group Meeting.

2. I are aware of the Office Action in the present application mailed December 8, 2006 in which Examiner Jeffery L. Williams maintained "Minutes of IPSEC Working Group Meeting" (hereinafter, "IPSEC Minutes") in view of "Fragmentation Considered Harmful" to Kent as a basis for rejecting independent claim 1 of the present application under 35 USC § 103(a).

3. A am aware of an Office Action Response and an Amendment in the present application being filed in response to the Office Action and that this declaration is attached to the Amendment as part of Exhibit 4.

4. I am aware that any material derived from the inventors listed in the present application may not be used as prior art against the invention and patenting thereof, unless that material was publicly disclosed more than a year before the filing of the patent application. I am also aware that to show that material in the prior art derived from the inventors objective evidence must be provided as to the same. I hereby aver that the subject matter that I provided in the "Minutes of IPSEC Working Group Meeting" reference derived from the inventors.

5. Examiner Williams has cited a portion from the IPSEC Minutes attributed to me, William Dixon. (Exhibit 2 at Page 2). The presentation I gave at the IPSEC meeting is attached as Exhibit 3. At the time of the IPSEC Minutes recordation, I was an employee of Microsoft and worked closely with the inventors, Mr. Brian Swander and Mr. Christian Huitema. In my interactions with the Microsoft product team including Mr. Brian Swander and Mr. Christian

Huitema, I became aware of all the information regarding IKE fragmentation presented by me in the IPSEC Minutes. Therefore, all disclosure cited by Examiner Williams in the IPSEC Minutes derived from the inventors.

6. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that statements are made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such false statements may jeopardize the validity of the application or any patent issued thereon.

Date March 21, 2007

William Dixon